Ph.D. DISSERTATION

Road Map Generation and Vehicle Localization for Automated Driving

무인자율주행을 위한 도로 지도 생성 및 측위

권 기 풍

FEBRUARY 2016

DEPARTMENT OF ELECTRICAL ENGINEERING AND
COMPUTER SCIENCE
GRADUATE SCHOOL
SEOUL NATIONAL UNIVERSITY

Road Map Generation and Vehicle Localization for Automated Driving

무인자율주행을 위한 도로 지도 생성 및 측위

지도교수 서 승 우

이 논문을 공학박사 학위논문으로 제출함

FEBRUARY 2016

서울대학교 대학원

전기 컴퓨터 공학부

권 기 풍

권 기 풍의 공학박사 학위논문을 인준함

FEBRUARY 2016

위 원 장	이 경 수	
부위원장	서 승 우	
위 원	김 현 진	
위 원	김 성 우	
위 원	정 한 유	

Abstract

This dissertation aims to present precise and cost-efficient mapping and localization algorithms for autonomous vehicles. Mapping and localization are ones of the key components in autonomous vehicles. The major concern for mapping and localization research is maximizing the accuracy and precision of the systems while minimizing the cost. For this goal, this dissertation proposes a road map generation system to create a precise and efficient lane-level road map, and a localization system based on the proposed road map and affordable sensors.

In chapter 2, the road map generation system is presented. The road map generation system integrates a 3D LIDAR data and high-precision vehicle positioning system to acquire accurate road geometry data. Acquired road geometry data is represented as sets of piecewise polynomial curves in order to increase the storage efficiency and the usability. From extensive experiments using a real urban and highway road data, it is verified that the proposed road map generation system generates a road map that is accurate and more efficient than previous road maps in terms of the storage efficiency and usability.

In chapter 3, the localization system is presented. The localization system targets an environment that the localization is difficult due to the lack of feature information for localization. The proposed system integrates the lane-level road map presented in chapter 2, and various low-cost sensors for accurate and cost-effective vehicle localization. A measurement ambiguity problem due to the use of low-cost sensors and poor feature information was presented, and a probabilistic measurement association-

based particle filter is proposed to resolve the measurement ambiguity problem. Experimental results using a real highway road data is presented to verify the accuracy and reliability of the localization system.

In chapter 4, an application of the accurate vehicle localization system is presented. It is demonstrated that sharing of accurate position information among vehicles can improve the traffic flow and suppress the traffic jam effectively. The effect of the position information sharing is evaluated based on numerical experiments. For this, a traffic model is proposed by extending conventional SOV traffic model. The numerical experiments show that the traffic flow is increased based on accurate vehicle localization and information sharing among vehicles.

Keywords: Road map, roadway map, piecewise polynomial, vehicle localization

Student Number: 2009-20753

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and Motivations

Significant efforts have been made in academia and in industry to develop more intelligent transportation systems (ITSs) that can provide a higher level of safety and convenience. Over the past ten years, starting with the development of various ADASs (advanced driver assistance systems), development of autonomous vehicles has been actively conducted, which is the eventual goal of the ITS research. In order for the automated driving to be realized, a lot of component functions should be supported including environmental perception, vehicle localization and motion planning, etc. Among these functions, vehicle localization is one of the most basic and important component for the autonomous vehicles, which is based on the many other autonomous vehicle functions. For example, vehicle path and motion planning start from accurate vehicle localization and various environmental perception algorithms benefit from the vehicle localization.

The vehicle localization can be defined as a function that provides accurate and precise location coordinates of the vehicle on a map. This definition contains a meaning that it should be preceded to create precise road maps for accurate vehicle localization, and then the localization is performed based on the information stored in the map. In this context, studies on the vehicle localization have been conducted in two aspects: 1) creating an accurate and useful road map (mapping), and 2) developing an accurate and efficient localization algorithm.

In fact, we are frequently experiencing the mapping and localization technologies in our daily lives through the vehicle navigation systems. The vehicle navigation systems have a road map and they provide the information about the vehicle location for drivers. However, conventional vehicle navigation systems just provide meter-level location information, thus they are inappropriate to use for autonomous vehicles. The inaccuracy of the vehicle navigation systems comes from the inaccuracies of both the mapping and localization methods. Generally, a vehicle navigation system uses a road map that only contains simplified road geometry information. It represents the road as a network that consists of nodes (crossways) and links (road sections) rather than expressing lane-level road geometry or more complex road feature information. Furthermore, the vehicle navigation systems use a low-cost GPS receiver for localization. Commonly, a low-cost GPS has a meter-level error and it can be increase to tens of meters according to the surrounding environment.

Actually, the vehicle localization problem gets easier with high-tech equipment such as RTK-GPS, high-end INS, high-end computing and storage systems, etc. Therefore, studies on mapping and vehicle localization have been conducted to reduce required cost while increasing the accuracy and precision. In this context, this dissertation aims to propose cost-effective and precise mapping and localization systems

for autonomous vehicles.

In chapter 2, a mapping system to generate a precise and cost-effective lane-level road map is proposed. It is shown that the map generated by the proposed system has enough accuracy to use for automated driving and highly efficient in terms of storage space and usability. In chapter 3, a localization system is proposed based on the lane-level road map and low-cost sensors including in-vehicle network sensors, low-cost GPS and monocular camera. It is shown that the proposed system provides accurate vehicle positioning information with low-cost sensors under an environment in which feature information for localization is poor. In chapter 4, an application of the accurate localization is presented. It is demonstrated that accurate vehicle localization and inter-vehicle communication can increase the traffic flow significantly.

1.2 Contributions and Outline of the Dissertation

1.2.1 Generation of a Precise and Efficient Lane-Level Road Map

Conventional road maps that are used for vehicle navigation systems or geographical information systems are insufficient to satisfy new requirements of intelligent vehicle systems such as automated driving. However, a highly precise map that satisfies the requirements for intelligent vehicle systems requires large amount of storage and thus has a low level of usability. In chapter 2, I propose a precise lane-level road map system that is suitable for automated driving and also conforms to the required storage efficiency and usability criteria. The proposed map building process consists of three steps: 1) data acquisition, 2) data processing, and 3) road modeling. The road data acquisition and processing system captures accurate 3D road geometry data by acquiring data with a mobile 3D laser scanner. The road geometry data is then refined to extract meta information, and in the road modeling system, the refined data

is represented as the set of polynomial-based spline curves to ensure storage efficiency and usability of the map. The proposed mapping system has been extensively tested and evaluated on a real urban road, and highway. The results of the experiment indicated that the proposed mapping system outperforms conventional ones in terms of storage efficiency and usability.

1.2.2 Accurate and Cost-Effective Vehicle Localization in Featureless Environments

There have been fewer studies on vision-based/map-aided vehicle localization for the environment where feature information for localization is poor, e.g., a highway environment. In chapter 3, I propose a precise and cost-effective vehicle localization system based on four cost-effective data sources including a low-cost GPS, in-vehicle network sensors, a monocular camera, and a high-precision lane-level road map for featureless environments. The combination of the low-cost sensor configuration and the poor feature information causes an ambiguity problem in the map-image matching process for localization. To handle this problem, a probabilistic measurement association-based particle filter is proposed. The proposed localization system has been evaluated on a real highway road. The results of the experiments demonstrate that the proposed localization system effectively solves the measurement ambiguity problem and provides accurate vehicle pose information for automated driving.

1.2.3 An Application of Precise Vehicle Localization: Traffic Flow Enhancement Through the Sharing of Accurate Position Information Among Vehicles

Traffic jam is a serious social problem in the view point of economy and environment. In chapter 4, I show that precise vehicle localization can be utilized to enhance the

traffic flow by sharing the position information among vehicles through a numerical experiment. The stochastic optimal velocity (SOV) traffic model that is well known to model the actual traffic flow correctly is used for the numerical experiment. I proposed an extended SOV (ESOV) model reflecting the effect of the position information sharing while retaining the generality of the conventional SOV model, and demonstrated that the sharing of accurate position information among vehicles can improve the traffic flow significantly using the ESOV model.

Chapter 2

Generation of a Precise and Efficient Lane-Level Road Map

It is necessary to have a precise lane-level road map to provide various autonomous driving vehicle applications, such as vehicle path or motion planning [1–4] and ego-vehicle localization [5–10]. In addition, various advanced driver assistance systems (ADASs) including lane keeping, lane change assist, and fuel management systems, benefit from having precise lane-level road maps [11].

From a technical point of view, a map for intelligent and autonomous vehicle systems has to fulfill the following three requirements: accuracy, storage efficiency, and usability. First, a road map should contain the geometry of all lanes with an accuracy at the centimeter-level as well as the 3D structure of roads in order to represent various types of roads, including sloped roads, overpasses, etc. Second, the geometry of the road should be expressed in a compact form such that the map can be downloaded and updated via wireless networks. Finally, the road geometry should be expressed in an application-friendly format. Specifically, low computational efforts

should be necessary to calculate the road geometry information, including coordinates, tangent angle or curvature from the road map to provide real-time operation.

Conventional road maps only provide macro-scale information for vehicle navigation systems or geographical information systems (GISs), so many studies have been carried out to produce a precise road map intended for use with intelligent vehicle systems. Bétaille *et al.* proposed an Emap structure for meso-scale lane-level driving assistance that utilizes GPS and Dead Reckoning (DR) [12]. However, the quality of the map is easily affected by the data acquisition process and is inadequate for autonomous driving purpose. A map for autonomous driving requires micro-scale precision, but such a precise map requires a large amount of storage space and has a low level of usability [13]. Jo *et al.* proposed a mapping system that can generate a storage-efficient map suitable for autonomous driving purposes without considering lanes, complex junctions and 3D traffic structures such as an overpass and a slope [14].

Few previous studies have considered the latter two requirements while providing centimeter-level functionality. In this chapter, I propose a road map generation system that considers the three aforementioned road map requirements altogether. The road map generation system is composed of three subsystems, including data acquisition, processing, and road modeling systems. The data acquisition and processing system acquires accurate road geometry data that fulfills the accuracy requirement, and I introduce a data acquisition system to obtain highly-accurate 3D road geometry by utilizing 3D mobile laser scanning (3D MLS) data. Then, the road geometry data that is obtained is refined to extract the corresponding meta information.

The main contribution of this chapter lies in the road modeling system. In the road modeling system, the road geometry data that is obtained is approximated and represented as sets of piecewise parametric polynomials in order to conform to the

latter two road map requirements. The piecewise polynomial is more effective in terms of usability than those previously used in other road models, including clothoid or B-spline, because various types of road geometry information such as the tangent angle and the curvature of the road can be calculated by conducting simple arithmetic operations. In addition, the piecewise polynomial form is suitable for local modification of the map, since the modification of the coefficients of a piecewise polynomial does not affect the shape of the other piecewise polynomials.

The main problem in representing the road as the sets of piecewise polynomials is the need to minimize the required data size, since the geometry data about the length of the tens-of-thousands kilometers of roads should be stored in a storage device of a vehicle. However, previous piecewise polynomial-based curve approximation algorithms are inappropriate for processing the significantly large amount of road geometry data, since these global optimization algorithms have exponential or $O(n^3)$ computational complexity [15, 16]. In this chapter, I propose an efficient road modeling algorithm that has $O(n)$ complexity. The proposed algorithm sub-optimally determines the number of piecewise-polynomials of the overall curve and its coefficients by converting the global approximation problem to a combination of small easy-to-solve approximation problems.

The proposed road map generation system has been extensively tested and evaluated on an urban road and a highway, and the results of the experiment show that the proposed map outperforms conventional maps in terms of the road map requirements. Our scheme can thus contribute to building a precise and efficient map for navigation systems that can be embedded in autonomous driving vehicles and updated via wireless networks.

The main contributions of this chapter can be summarized as follows:

- I propose a road map generation system to generate a precise and efficient lane-level road map for intelligent and autonomous vehicle systems.
- I propose an efficient curve approximation algorithm that represents the road as the minimum number of piecewise polynomials.
- The feasibility and practicality of the proposed map were evaluated by conducting extensive experiments on an urban road and a highway.

The remainder of this chapter is organized as follows. Section 2.1 presents related works. Section 2.2 presents the overall system architecture of the road map generation system. Section 2.3 introduces the road geometry data acquisition and processing system, and Section 2.4 presents the proposed road representation system. Section 2.5 provides the experimental results and Section 2.6 summarizes this chapter.

2.1 Related Works

The accuracy, storage efficiency and usability of a map all depend on the combination of data acquisition and road modeling methods that are used. The objective of data acquisition is to obtain accurate data to represent the actual geometry of the road, and the primary purpose of road modeling is to effectively represent the road geometry in terms of the storage efficiency and usability while maintaining a certain level of accuracy. Table 2.1 presents the data acquisition and road modeling methods used in previous representative studies and also describes the extent to which they meet the three road map requirements for intelligent vehicle systems. The vertical axis of the table represents the classification of the data acquisition methods based on the level of accuracy that the methods can meet, and the horizontal axis represents the classification of the road models based on the level of usability. The asterisk in

Table 2.1 Comparison to representative previous road map generation methods.

			Usability				
			N/A	Low	Mid.		High
		Road model / Data Acquis.	None	Polygon	Clothoid	B-spline	Set of piecewise polynomials
Accuracy	Meter	Aerial/satellite image		OSM[a] [17]			
	Decimeter	GPS+DR[b] (INS[c])			Bétaille, *et al.* [12]*	Jo, *et al.* [14]* .	
	Centimeter	GPS+DR (INS)+Camera			Ziegler, *et al.* [18]		
		GPS+DR (INS)+3D Lidar	Joshi, *et al.* [19]				Proposed*

*Methods where the maximization of storage efficiency is considered.

[a]Open street map, [b]Dead reckoning, [c]Inertial navigation system.

the table indicates that the marked method considers the maximization of storage efficiency. The table shows that no prior system has simultaneously achieved the three road map requirements for intelligent vehicle applications. A more detailed description about the previous approaches is presented in the rest of this section.

2.1.1 Acquisition of Road Geometry

Various approaches have attempted to acquire accurate road geometry data. For example, an aerial/satellite image-based approach has been extensively used for conventional digital maps and GISs. High-resolution aerial camera images have been acquired from a satellite or an aerial vehicle, to extract road geometries through manual work or through image processing means [20–22]. An image-based approach has an advantage in that we can obtain the road geometry for a larger region by processing a single image. However, the resolution of the aerial/satellite images is insufficient for extracting precise lane-level road geometry. Moreover, the elevation information of the roads cannot be acquired because the images contain no depth information. As a consequence, the accuracy of the image-based approach is limited to meter-level.

Many studies have adopted a probe vehicle-based approach to acquire a more accurate road geometry [11, 12, 14, 23–25]. In this approach, a probe vehicle equipped with various sensors explores roads and collects sensor data to obtain road geometry information. Of the various sensor configurations that are possible, kinematic GPS-based methods are the most widely used systems that include real-time kinematic (RTK) and post-processing kinematic (PPK) GPSs. In this method, the trajectory of a probe-vehicle driving along the centerline of a road (or lane) is recorded as road geometry data. The accuracy of the road geometry depends mainly on the posi-

tioning accuracy of the vehicle, so many previous studies have used algorithms that integrate GPS with other positioning systems, such as dead reckoning (DR) and inertial navigation system (INS), to improve the accuracy and reliability of the vehicle position [14, 26, 27].

Although the GPS-based methods are useful to obtain a greater degree of accuracy in the road geometry than image-based methods, there are some fundamental limitations to using this method. First, it is inefficient for the probe vehicle to capture information for a road multiple by as many times as the number of lanes. Such repetitive work is costly and time-consuming. The second problem is the so-called trajectory error. In a GPS-based method, it is desired for a probe vehicle to drive along the exact centerline of the lane. However, it is practically impossible for a human driver to control the vehicle with centimeter-level precision for an extended period of time. Therefore, discordance between the vehicle trajectory and the centerline of the lane is unavoidable. Thus the accuracy of using this method is limited to decimeter-level.

Various methods based on perception sensor have been introduced in recent years to tackle the problems of using GPS-based methods [18, 24, 28–31]. In these methods, road markings are detected and extracted by using perception sensors that capture road geometry data. Since the road geometry data is acquired directly from lane marking information, the above problems are fundamentally prevented. Various sensor configurations have been used to this end, including a monocular camera [24], stereo-camera [18], and 2D or 3D Lidar [19, 29–31]. Camera-based methods have an advantage in that the cost of the sensor can be reduced; however, it is difficult to extract precise 3D road geometry because of the inherent limitation of a camera, which is that the information is represented in a 2D plane. For this reason, the 3D Lidar is the most appropriate way to acquire centimeter-level road geometry since it

can provide accurate 3D information about roads. Some studies have extracted road geometry from 3D Lidar data. In [29], an algorithm was proposed to extract road geometry from 3D Lidar data; however, the algorithm focused on extracting road regions rather than lane-level road geometry. In [30] and [31], the extraction of lane markings from 3D Lidar data was considered, but, the clustering among lanes was not addressed. Joshi *et al.* proposed a particle filter-based method to extract lane-level geometry data and clustering among the lanes using 3D Lidar data [19]. However, that method extracts the centerline of lanes rather than the lane marking points, so the shape of the lane markings (e.g., the dashed line) is ignored, which can be useful for intelligent vehicle applications such as vehicle localization. In this chapter, I propose a 3D Lidar data processing algorithm to extract and cluster the lane marking points.

2.1.2 Modeling of Road Geometry

It is important to have the appropriate road geometry representation to ensure storage efficiency and usability along with map accuracy. Various road geometry models have been previously proposed [11, 12, 14, 24, 32, 33], but the previous models have not considered the three requirements simultaneously. For example, polygons are widely used in conventional digital road maps and various intelligent vehicle applications to represent the road geometry due to their simplicity [17, 34–37]. However, since a polygon cannot express curved roads with precision, too many line segments become necessary to express the roads with a large curvature in order to achieve centimeter-level accuracy. Therefore, the accuracy and storage efficiency requirements cannot be simultaneously satisfied. In addition, direct exaction of road geometry information cannot be provided, including the tangent angle and curvature, and thus the usability

requirement is also not satisfied.

Various mathematical curve models have been proposed to provide a more accurate and efficient road representation. For example, a clothoid is the best way to represent road geometry since a road is traditionally designed by a set of clothoids [12,38]. Bétaille *et al.* proposed a clothoid-based road modeling algorithm that satisfies the accuracy and storage efficiency requirements [12]. In this algorithm, the road geometry is represented as a set of the minimum number of clothoid curves while also satisfying a preset accuracy constraint. However, a clothoid is inadequate for use with intelligent vehicle applications since it includes transcendental functions. Thus the calculations for the information required for advanced vehicle applications, such as distance and relative angle between an ego-vehicle and a lane in the map, are computationally intense. As a result, the clothoid does not satisfy the usability requirement. Moreover, a clothoid cannot express the 3D road geometry by itself since the clothoid is only defined on a plane.

An alternative is to use a spline curve. Many studies have used various types of spline curves to represent the road geometry [11, 14, 24, 32, 33]. For example, a cubic B-spline is a representative method for road modeling that has an advantage in that local modifications of the curve do not affect to the entire shape of the curve. This is an important characteristic for road map maintainability, and in addition, many efficient B-spline curve approximation algorithms that have been developed for computer aided design (CAD) make it easy to use B-spline curves for road modeling. For example, in literature [14], a B-spline-based road modeling algorithm was proposed based on a dominant points-based B-spline curve fitting algorithm developed for computer-aided design (CAD) [39]. The control point of the B-spline is sub-optimally determined according to a preset accuracy constraint in order to simultaneously adhere to the

accuracy and storage-efficiency requirements. However, it is difficult to extract the information on the geometry of the road, such as the tangent angle and curvature by using a B-spline curve because the functions of the B-spline are not intuitive, and it is thus difficult to calculate the first and second derivatives of the functions [40]. Therefore, B-splines also do not satisfy the usability constraint.

In this dissertation, I use the most intuitive spline curve form where a curve segment is expressed as a piecewise cubic polynomial. Since a polynomial can be used to conveniently calculate the derivatives, we can extract the tangent angle and curvature of the road from the proposed spline curve by using simple arithmetic operations. Piecewise polynomial-based curve approximation has not been studied as much as other spline curves such as the B-spline. Moreover, previous piecewise polynomial approximation algorithms are not appropriate for approximating significantly large amount of road geometry data since the previous algorithms have exponential or $O(n^3)$ computational complexity [15, 16]. One solution is to approximate the data as a B-spline curve first and to convert the B-spine curve to a set of piecewise polynomials [41]. However, this approach requires a great number of piecewise polynomials than directly approximating the data as a set of piecewise polynomials as will be shown in Section 2.5.3. Thus, this chapter proposes an efficient piecewise polynomial curve approximation algorithm that processes a significantly large amount of road geometry data and sub-optimally determines the number of piecewise polynomials and their coefficients with $O(n)$ computational complexity.

2.2 Overall System Architecture

The overall road map generation system is composed of three subsystems: a data acquisition, a data processing and a road modeling subsystems (see Fig. 2.1). The data

acquisition and processing subsystems acquire accurate and reliable road geometry data by using various sensors during collection on a probe vehicle equipped with a GPS+INS vehicle positioning system and a 3D laser scanner. Then, accurate road geometry data is extracted by integrating the sensor data in the data processing system.

In this dissertation, two different types of road geometry are acquired depending on the road type. For a road in which lane markings exist, the Cartesian coordinates of the lane markings are obtained as the road geometry data from the 3D laser scanning data (see the black circles in Fig. 2.1). For a road in which lane markings do not exist, such as when going off-road, the trajectory of a probe vehicle driving along centerline of a road is used in a manner similar to the previous probe vehicle-based approach (see the white squares in Fig. 2.1). For convenience, I have name the road in which lane markings exist as a *type I* road and the other as a *type II* road in the rest of this chapter.

Both types of road geometry data obtained from the data acquisition and processing systems are represented by a large number of points. However, this type of geometry representation is inadequate for a road map in terms of the storage efficiency. Moreover, the point representation makes it difficult to extract the required road geometry information, such as tangent angle and the curvature of a road. To solve these problems, I have applied a mathematical modeling technique to the road geometry point data in order to increase its storage efficiency and usability. The road modeling system approximates a set of geometry point data that correspond to a road line as a cubic spline curve consisting of sequential parametric piecewise cubic polynomials curves (see the right bottom of Fig. 2.1). A *sequential approximation algorithm* is proposed to efficiently approximate the curve.

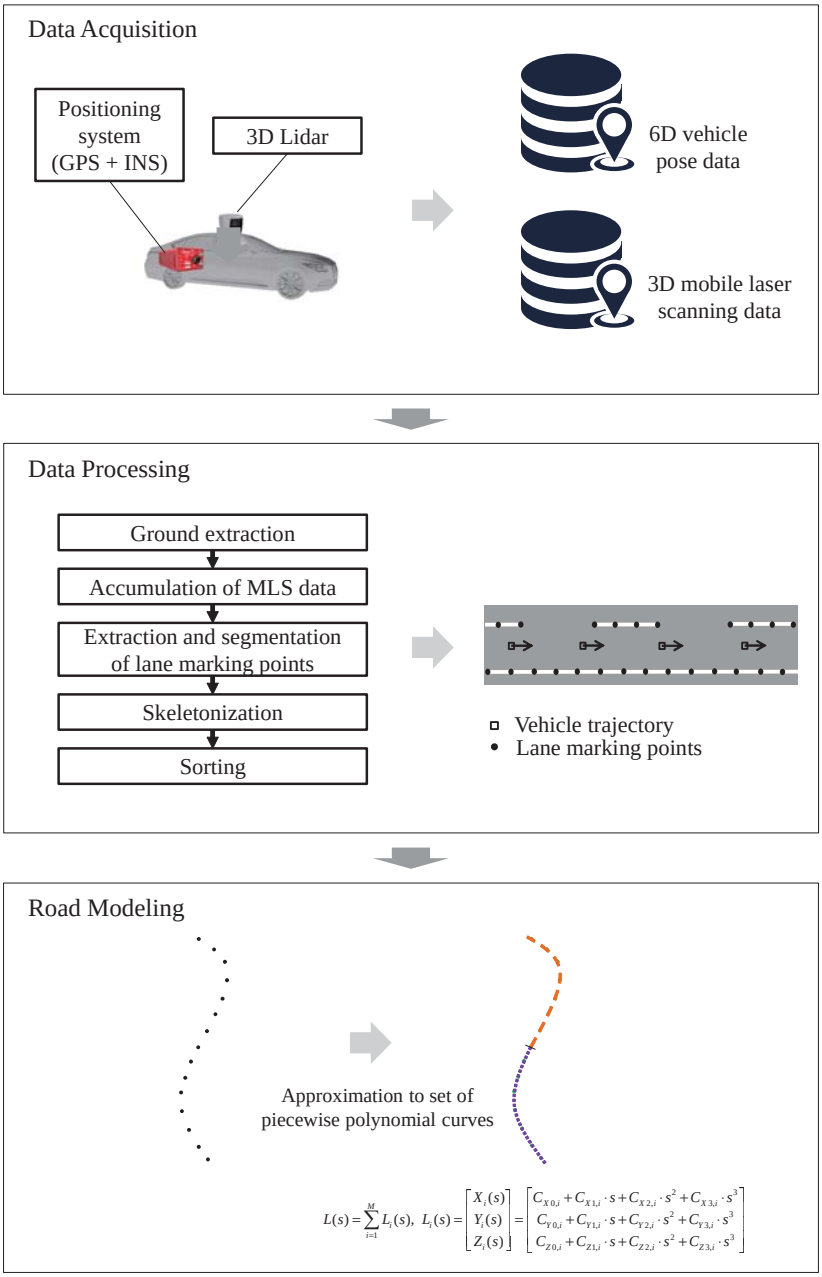


Figure 2.1 Overall System Architecture.

2.3 Road Geometry Data Acquisition and Processing

In the data acquisition, the required raw sensor data is collected by using a probe vehicle equipped with a positioning system and a 3D Lidar, and the geometry data

of the lanes is then obtained from raw data in the data processing step. Detailed description of the two steps is presented as follows.

2.3.1 Data Acquisition

During the data acquisition step, two kinds of data are collected by the probe vehicle: 6D vehicle pose data and 3D MLS data. The vehicle pose data includes the accurate 3D global position and 3D attitude (yaw, pitch and roll) with respect to the trajectory of the probe vehicle. In order to obtain the accurate vehicle pose data, a high-precision vehicle positioning system is recommended. In this dissertation, I use a sensor fusion system that integrates a RTK-GPS and a high-precision INS. The 3D MLS data is obtained by correcting the raw 3D laser scanning data using the INS data. When a vehicle moves, the measurement origin of the laser scanner is changed; thus the laser points are twisted [42]. To resolve this problem, a preprocessor untwists the points using the vehicle motion sensor measurements from the INS.

For *type II* roads, the vehicle pose data is simply used as the road geometry data, and for *type I* roads, MLS data is additionally used. The rest of this section presents the data processing algorithm that is used to obtain accurate road geometry data from the raw data. Note that this dissertation focuses more on a *type I* road since the road geometry for *type II* roads can be obtained by using the previous probe vehicle-based methods.

2.3.2 Data Processing

The road geometry data for a *type I* road can be obtained by integrating 3D MLS data with the vehicle pose data. Fig. 2.2 depicts the overall procedure for the data processing step. First, ground extraction is applied to each frame of the MLS data

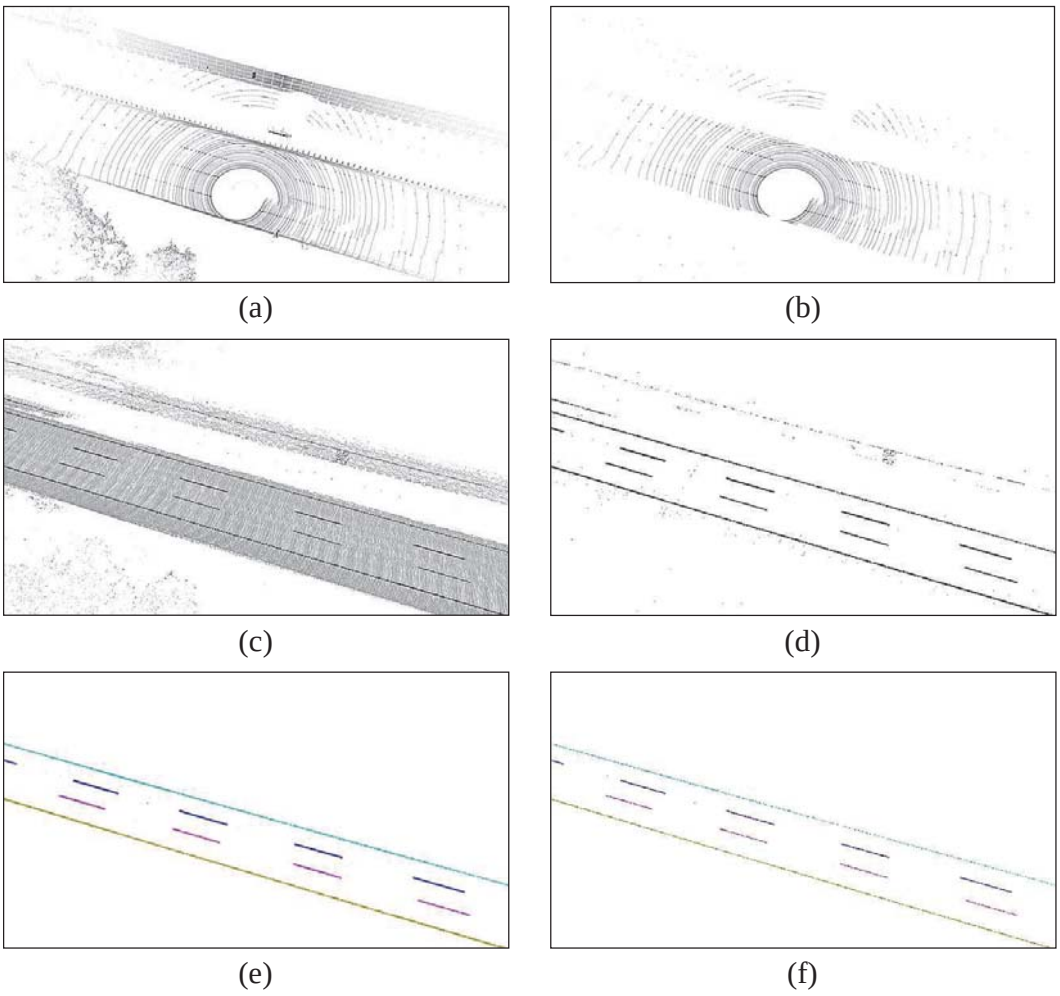


Figure 2.2 Overall data processing procedure: (a) raw MLS data, (b) ground extraction result, (c) result of data accumulation, (d) result of intensity thresholding, (e) result of lane marking point extraction and clustering, and (f) skeletonization result.

to remove unnecessary points from the MLS data, and then, the ground points for every frame of the MLS data are accumulated on a global coordinate system by using 6D vehicle pose data synchronized with MLS data. The accumulated ground points with a high intensity are then extracted as candidate points that correspond to lane markings, and the exact lane marking points are extracted and clustered by

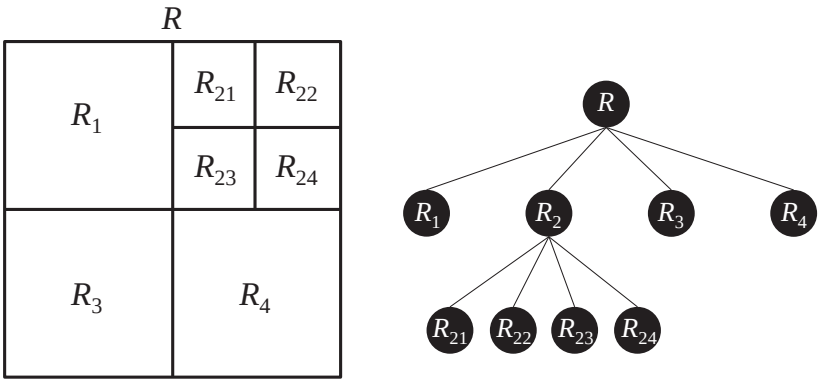


Figure 2.3 Quadtree partition structure which is used for ground extraction.

using more complex algorithms. Finally, the desired geometry data is obtained by skeletonizing the dense point data.

Ground extraction

Since the lane marking points on the ground surface are the only points of interest in this dissertation, removing off-ground points from each MLS is helpful for later processes in terms of the processing time and performance. Ground extraction is carried out according to an adaptive quadtree partition structure, as shown in Fig. 2.3. Initially, the MLS data points are divided into multiple cells with a size of w_{init}, and then, each cell is tested to identify whether the cell is a ground cell or not. If it is not certain whether a cell is a ground cell or not, the cell is divided into four child cells, and the test is applied to all child cells. This process is then continued until all the cells have been identified. The test to identify this type of cell is conducted as follows.

First I assume that there is at least one cell among the initial cells that has already been identified as a ground cell. This assumption is valid since it is certain that the

cells near the probe vehicle are ground cells. Then, the test is performed starting from the unidentified cells neighboring the identified cells. Let an unidentified cell be R_{UI}. Then the roughness of the cell is calculated to determine whether the cell is a ground cell or not by

$$\Delta z_{cell} = z_{\max}(R_{UI}) - z_{\min}(R_{UI}), \tag{2.1}$$

where $z_{\max}(\cdot)$ and $z_{\min}(\cdot)$ refer to the z coordinate value of the highest points and the lowest point in the cell, respectively. If the roughness value of the cell is larger than a given threshold, this means that the cell has to include the off-ground points. In this case, the cell is unidentified, and thus is divided into four child cells again. On the other hand, if the roughness value of the cell is smaller than the given threshold, it is then considered to be a candidate ground cell. For a candidate ground cell, the identification whether the cell is actually a ground cell is performed according to the following test:

$$|z_{\text{mean}}(R_{\text{UI}}) - z_{\text{mean}}(R_{\text{I,Nbh}})| < \lambda_{\text{GND}} \tag{2.2}$$

where $R_{\text{I,Nbh}}$ denotes the ground cell that has already been identified among the neighborhood of the test cell R_{UI}, and $z_{\text{mean}}(\cdot)$ refers to the mean value of heights of the points. If the test result is true, the height difference between the test cell and the neighbor ground cell is small, and therefore, the cell is identified as a ground cell. If the test result is false, then the test cell is identified as an off-ground cell. As a consequence, a cell is categorized into three states after this process: a ground cell, an off-ground cell or an unidentified cell. For the unidentified cells, the above process is repeated until all cells have been identified. Fig. 2.2b shows an example of the ground extraction.

Accumulation of MLS data

Every frame of the MLS data from which off-ground points have been removed is accumulated on a global coordinate system by using synchronized vehicle pose data. Basically, the MLS data is represented on a vehicle body coordinate system, and therefore, a rigid body transformation is applied to accumulate these in a global coordinate system. The rigid body transformation is simply defined by a transformation matrix, e.g.,

$$T_H = \begin{bmatrix} \cos\alpha\cos\beta & \cos\alpha\sin\beta\sin\gamma - \sin\alpha\cos\gamma & \cos\alpha\sin\beta\cos\gamma + \sin\alpha\sin\gamma & X_t \\ \sin\alpha\cos\beta & \sin\alpha\cos\beta\sin\gamma + \cos\alpha\cos\gamma & \sin\alpha\cos\beta\cos\gamma - \sin\alpha\sin\gamma & Y_t \\ -\sin\beta & \cos\beta\sin\gamma & \cos\beta\cos\gamma & Z_t \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{2.3}$$

where α, β and γ are the Euler angles of the vehicle body, respectively, and (X_t, Y_t, Z_t) represents the position of the vehicle in a global coordinate system. The MLS data is accumulated on the global coordinate system by the rigid body transformation, and finally, dense point cloud data representing the ground region is obtained, as shown in Fig. 2.2(c).

Extraction and Clustering of Lane Marking Points

Now, the points corresponding to the lane markings are extracted and are clustered among the points that belong to the same road line. First, rough lane marking points are extracted by using simple intensity-based thresholding, i.e., the points with a higher intensity than the threshold is extracted. I assume that an appropriate static intensity threshold is given for a road, since the intensity value of the Lidar is robust to external factors such as illumination. As we can see in Fig. 2.2(d), the majority of

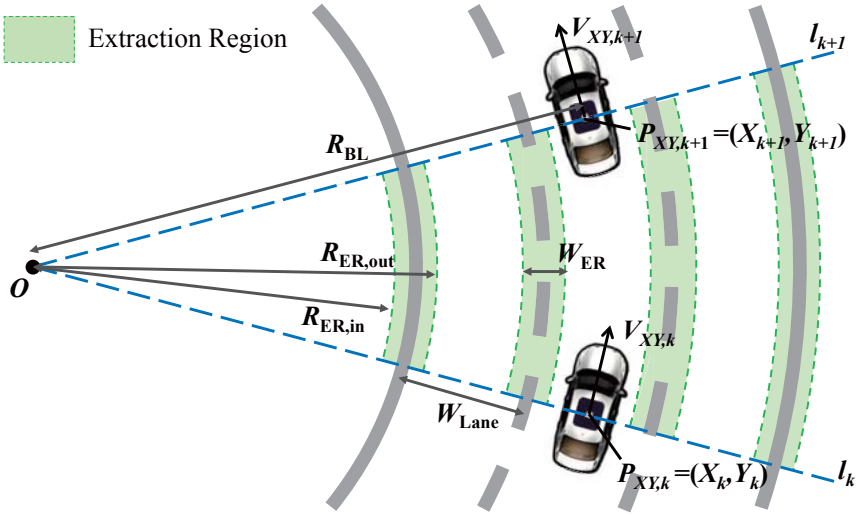


Figure 2.4 Lane marking point extraction and clustering.

non-lane marking points are effectively filtered out just through the simple intensity thresholding method. However, there are still exist outliers, and in addition, the points are not clustered by the road line. Therefore, an additional algorithm is applied to remove the remaining outliers and to cluster the points by the road line, and since it is difficult and inefficient to process all points at once, I first vertically divide the point cloud into a set of data blocks by using the vehicle pose data in the XY plane. A block is then defined by two lines, named a split line, where the kth split line, l_k, is a line that is orthogonal to the yaw of the kth vehicle pose and the passing vehicle position $P_{XY,k}$ (see the coarse dashed lines in Fig. 2.4). With the split lines, all points are divided into $N_p - 1$ blocks, where N_p is the number for the vehicle pose data (number of split lines). These data blocks are then processed separately to extract the lane marking points as follows.

The algorithm starts by finding an intersection for two split lines, l_k and l_{k+1}, and the radius R_{BL} with respect to the intersection point. Note that I assume that

the probe vehicle drives along the centerline of a lane as much as possible, and the vehicle trajectory between time k and $k+1$ can be modeled by an arc since the time gap is small enough (10 ms in this dissertation). After that, the lane marking point extraction regions in which the lane marking points are likely to exist are set using some prior information of the road, e.g, the number of lanes, N_{L}; width of a lane, W_{L}; and the lane number in which the probe vehicle drove, n_{drv} ($n_{\mathrm{drv}} = 1$ for the leftmost lane). This information can be obtained using conventional maps, such as Open Street Map (OSM) [17], or by using camera-based algorithms [43–45]. There is an exceptional case where the yaw angles of two consecutive vehicle pose data are exactly the same; thus, the intersection point of two split lines does not exist. In this case, the radius R_{BL} is set to a value that is large enough (e.g., 3000 m), and the origin O is set to a point so that it satisfies $\|P_{XY,k} - O\| = \|P_{XY,k+1} - O\| = R_{\mathrm{BL}}$. This trick is valid since the radius is large enough compared to the length of the arc between points $P_{XY,k}$ and $P_{XY,k+1}$ (about 2.7 m where the speed of the vehicle is 100 km/h and the pose data is acquired at a 10 Hz data rate) As shown in Fig. 2.4, the extraction region is bounded by two inner and outer arcs, and the radiuses of the arcs are determined by R_{BL}, N_{L} and W_{L} as follows:

$$R_{\mathrm{ER},i} = \{R_{\mathrm{ER},i,\mathrm{in}}, R_{\mathrm{ER},i,\mathrm{out}}\} = R_{\mathrm{BL}} + (2i - 2n_{\mathrm{drv}} - 1)\frac{W_L}{2} \pm \frac{W_{ER}}{2}, \tag{2.4}$$

where i denotes the index of the road lines from the left (e.g., $i = 1$ for the leftmost road line), and the width of the extraction region, W_{ER}, is determined by considering the uncertainty degree with respect to the error between the expected location of the lane marking points and the true location of these. Although the larger value for W_{ER} may decrease the false negative rate, more non-lane marking points may be extracted. On the other hand, a smaller W_{ER} value may exclude non-lane marking points as well, which would decrease the true positive rate. Therefore, it is important to carefully

determine W_{ER} value to ensure reliable road marking point extraction. Finally, the points inside of the extraction region are extracted as the lane marking points, and the points belonging to the same region are grouped together. After finishing the extraction procedure, the points belonging to the same region are naturally grouped, and thus the lane marking points are clustered into several clusters that correspond to each of the road lines.

Skeletonization

In the previous steps, the lane marking points were obtained as shown in Fig. 2.2(e). However, since the lane marking points describe thick lane markings at this stage, it is difficult to represent lane markings as lines by using this set of lane marking points. Skeletonization is applied to the lane marking points to represent the lane markings as lines, as shown in Fig. 2.2(f). Skeletonization is a technique that is used to simplify and abstract a volumetric object to a line shape [46]. In this dissertation, the skeletonization is applied to lane marking points for extracting the points that correspond to the centerlines of the lane markings.

Sorting

For the final process, the lane marking points within each cluster are sorted and sequentially indexed from the closest point to the farthest point with respect to the starting point of the road. However, applying the sorting algorithm to all of the points is time-consuming. Fortunately, the points were divided into multiple blocks in the lane marking point extraction process. Since the blocks are sequentially arranged from the starting point of the road, we can sort all of the points by sorting the points in each block separately and concatenating the blocks of sorted points. In this dissertation,

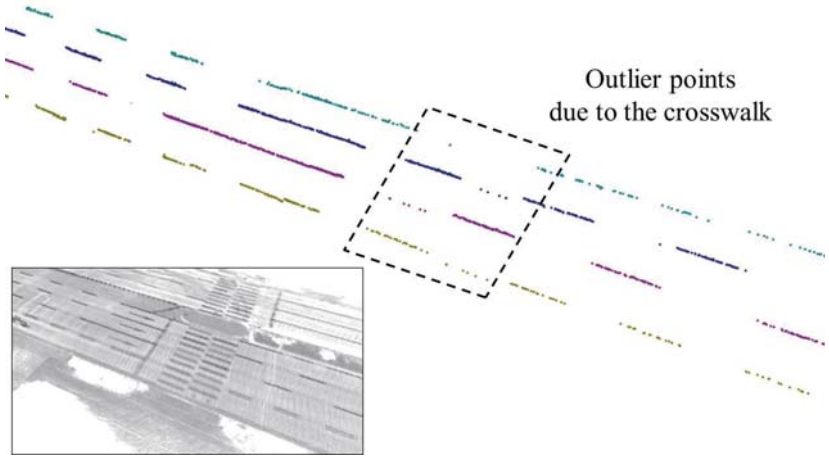


Figure 2.5 Example of outlier points.

the bubble sorting algorithm is used in a block. As a result of the data sorting, the geometry data of the road with respect to the road lines is represented as sets of sequential points as follows:

$$\begin{aligned} \mathcal{G} &= \{\mathcal{G}_i | i = 1, ..., N_L + 1\} \\ \mathcal{G}_i &= \{G_n = (X, Y, Z)_n | n = 1, ..., N_i\}, \end{aligned} \tag{2.5}$$

where $\mathcal{G}_i$ is a set of sequential lane marking points corresponding to the ith road line, N_L is the number of lanes, G_n is the nth lane marking point, and N_i is the total number of lane marking points in $\mathcal{G}_i$.

2.3.3 Outlier Problem

The MLS-based lane marking extraction algorithm is susceptible to outliers that can be extracted with true lane marking points. As shown in Fig. 2.5, the road markings that exist outside of the extraction region such as arrows are effectively excluded by the extraction algorithm. However, the road markings that exist inside the extraction region (road markings that are well aligned with the lane markings) are extracted

with the lane markings. These outliers may prevent the extracted road geometry data points from accurately expressing the true geometry of the road. This problem is supplemented by the road modeling algorithm that will be proposed in a later section. The proposed road modeling algorithm was designed to accurately model the true geometry of the road even though there exist outliers in the data.

2.4 Road Modeling

The road geometry data $\mathcal{G}$ that is represented by a set of points is not suitable for use as a road map since it consumes a large amount of storage space and the road geometry information is difficult to extract from the data. In this section, I propose a road modeling algorithm that increases the storage efficiency and usability of a road map by representing each set of points, $\mathcal{G}_i$, corresponding to a road line, as a mathematical curve defined as

$$L(s) = \sum_{m=1}^{M} L_m(s), \tag{2.6}$$

where, the subscript i that refers to the index of the road line was omitted for convenience, and

$$L_m(s) = \begin{bmatrix} X_m(s) \\ Y_m(s) \\ Z_m(s) \end{bmatrix} = \begin{bmatrix} \sum_{p=0}^{3} C_{Xp,m} \cdot (s - s_m)^p \\ \sum_{p=0}^{3} C_{Yp,m} \cdot (s - s_m)^p \\ \sum_{p=0}^{3} C_{Zp,m} \cdot (s - s_m)^p \end{bmatrix}, \; for\, s \in [s_m, s_{m+1}). \tag{2.7}$$

The curve $L(s)$ is a cubic spline curve that is composed of a finite number of sequentially connected piecewise polynomial curves. Each piecewise polynomial curve is parameterized according to $s \in [s_i, s_{i+1}]$ and consists of three cubic polynomials

that represent the X, Y and Z coordinates, respectively. The advantage of representing the road geometry data points as a spline curve is obvious since the number points can be expressed as only one parameter value and twelve coefficients of a piecewise polynomial, i.e., s_m, $C_{X0,m} \sim C_{X3,m}$, $C_{Y0,m} \sim C_{Y3,m}$ and $C_{Z0,m} \sim C_{Z3,m}$.

The problem lies in expressing the point data in the spline curve form, that is, how to determine the number of piecewise polynomial curves and coefficients of the curves. There are three main considerations for this problem: storage efficiency, accuracy and outlier handling. First, the number of piecewise polynomial curves that compose the spline curve should be minimized to maximize the storage efficiency. However, there is a tradeoff between the number of piecewise polynomials and the accuracy, and in general, as we use a greater number of polynomial curves, the spline curves can express the point data with higher accuracy. Therefore, the problem is to minimize the number of piecewise polynomial curves while satisfying predefined accuracy constraint. Meanwhile, outliers included in the point data negatively impact the accuracy of the curve approximation. Therefore, the goal of the road modeling system is to find the optimal M, the number of piecewise polynomial curves, $\{s_m | m = 1, ..., M\}$, the set of parameters that indicate the point where the piecewise polynomial curves are divided, and the coefficients of the curves that take the accuracy constraint and outlier points into account.

However, it is very difficult to solve this problem from a global optimization point of view since we need to consider the number of piecewise polynomials, the parameter points at which the curve is divided, and the coefficients of the polynomials altogether. In addition, the accuracy constraint and the outliers should also be considered. Therefore, I consider the original problem as a combination of smaller problems that can be solved relatively easily, and I thus propose the *sequential approximation algorithm*

to efficiently find the near-optimal solution. In the rest of this section, I describe the details of the sequential approximation algorithm.

2.4.1 Overview of the sequential approximation algorithm

As mentioned above, the basic idea to solve the global optimization problem is to convert the problem into a combination of multiple simpler problems. The problem conversion process is based on the intuition that minimizing the number of piecewise polynomials that model the overall road geometry data is equivalent to maximizing the number of data points that each piecewise polynomial can express. I thus propose the sequential approximation algorithm, which can be summarized as below:

i. A set of road geometry point, $\mathcal{G} = \{G_n = (X, Y, Z)_n | n = 1, ..., N\}$, is given. Note that index i of the point set $\mathcal{G}_i$ is omitted for convenience.

ii. An initial polynomial curve is approximated by using the first few points. Here, the starting point of the curve is fixed for the first data point.

iii. Starting from the second point, a new point is sequentially added at every step, and the coefficients of the polynomial curve are corrected by using a Kalman filter (KF)-based algorithm.

iv. Whenever the polynomial curve is corrected, the accuracy of the curve is verified.

v. If the accuracy is higher than a threshold, step iii) is repeated. If not, adding new data points stops, and the coefficients of the polynomial curve that do not violate the accuracy criterion are saved.

vi. The remaining data points are used to repeat, ii) - v) until the last data point is has been reached.

As can be seen from the above description of the algorithm, the overall algorithm is composed of multiple sequential approximation processes. In an approximation process, a piecewise polynomial curve is generated to expresses as many points as possible within the accuracy bound. This approach is used to sub-optimally determine the set of break parameters, $\{s_m | i = m, ..., M\}$, and the coefficients of the curves. In the rest of this section, a detailed explanation of the sequential approximation algorithm is presented. First, I describe the detailed curve approximation process, and then present the detailed curve transition criterion and algorithm. The overall procedure for the algorithm is described in Algorithm 1.

2.4.2 Approximation Process

I use a Kalman filter, a recursive Bayesian estimation approach to approximate the given points to a cubic polynomial curve. The data points are regarded as observations for the estimation system, and the coefficients of the polynomial curve are regarded as a system state that we want to estimate. The reason to use a stochastic method instead of deterministic methods, such as spline interpolations, is that the data points contain stochastic errors even though the data has been refined in the data acquisition and processing systems. In addition, outliers included in the data should also be handled, and with proper parameter settings, a Bayesian filter can be designed to be robust against such outliers.

System Model

The system state vector x_k and the measurement vector y_k are defined by

$$\begin{aligned} x_k &= (C_{X1,k}, C_{X2,k}, C_{X3,k}, C_{Y1,k}, C_{Y2,k}, C_{Y3,k}, C_{Z1,k}, C_{Z2,k}, C_{Z3,k})^{\mathrm{T}}, \\ y_k &= (X_{y,k}, Y_{y,k}, Z_{y,k})^{\mathrm{T}}. \end{aligned} \tag{2.8}$$

Algorithm 1: Sequential Approximation Algorithm

input : $\mathcal{G}_i = \{G_{i,n} = \{X, Y, Z\}_{i,n} | n = 1, ..., N_i\}$

output: $L_i = \{L_{i,m} | m = 1, ..., M_i\}$

$m \leftarrow 1$; $k \leftarrow 1$; $s_k \leftarrow 0$

$[C_{0,m},\, x_{m,k|k},\, P_{m,k|k}] \leftarrow$ `Initialization`$(\mathcal{G}_i,\, k)$;

$[x_{m,k+1|k},\, P_{m,k+1|k}] \leftarrow$ `Prediction`$(x_{m,k|k},\, P_{m,k|k})$;

for $k = 2$ **to** N_i **do**

  $s_k \leftarrow s_{k-1}+$`DistanceBetweenPoints`$(G_{i,k-1},\, G_{i,k})$;

  if `OutlierCheck`$(x_{m,k+1|k},\, P_{m,k+1|k},\, G_{i,k})$ *false* **then**

    $[x_{m,k|k}, P_{m,k|k}] \leftarrow$ `Update`$(x_{m,k+1|k},\, P_{m,k+1|k},\, u_k,\, \mathcal{G}_{i,k})$;

  else

    $x_{m,k|k} = x_{m,k|k-1}$;

    $P_{m,k|k} = P_{m,k|k-1}$;

  end

  if (`CurveTransitionTrigger`$(\mathcal{G}_i,\, C_{0,m},\, x_{m,k|k})$) or $(k = N_i)$ **then**

    $k \leftarrow k - N_{\text{buff}}$;

    $L_{i,m} \leftarrow \{C_{0,m},\, x_{m,k|k}\}$;

    $m \leftarrow m + 1$;

    $[C_{0,m},\, x_{m,k|k},\, P_{m,k|k}] \leftarrow$ `Initialization`$(\mathcal{G}_i,\, k)$;

  end

  $[x_{m,k+1|k},\, P_{m,k+1|k}] \leftarrow$ `Prediction`$(x_{m,k|k},\, P_{m,k|k})$;

end

$L_i \leftarrow$ `ArcLengthParameterization`(L_i);

/*$C_{0,m} = \{C_{X0,m}, C_{Y0,m}, C_{Z0,m}\}$ */

Note that the constant terms of the polynomials are excluded from the system state vector to ensure the geometrical continuity between neighbouring piecewise curves. However, I do not regulate both C_1- and C_2-*continuity* since there can be points where the tangent or the curvature are discontinuous in real roads.

Based on the state and the measurement vector definitions, I establish linear prediction and measurement models that are suitable for Kalman filtering. The prediction model is defined as

$$\begin{aligned} & x_k = x_{k-1} + w_{k-1},\ w_{k-1} \sim N(0, Q_{k-1}) \\ \Rightarrow & \begin{bmatrix} \mathbf{C}_{X,k} \\ \mathbf{C}_{Y,k} \\ \mathbf{C}_{Z,k} \end{bmatrix} = \begin{bmatrix} \mathbf{C}_{X,k-1} \\ \mathbf{C}_{Y,k-1} \\ \mathbf{C}_{Z,k-1} \end{bmatrix} + w_{k-1}, \end{aligned} \tag{2.9}$$

and the measurement model is defined as

$$\begin{aligned} y_k &= H_k \cdot x_k + u_k + v_k,\ v_k \sim N(0, R_k) \\ y_k &= \begin{bmatrix} X_{\mathrm{y},k} \\ Y_{\mathrm{y},k} \\ Z_{\mathrm{y},k} \end{bmatrix} = \begin{bmatrix} C_{X0} + C_{X1,k} \cdot s_k + C_{X2,k} \cdot s_k^2 + C_{X3,k} \cdot s_k^3 \\ C_{Y0} + C_{Y1,k} \cdot s_k + C_{Y2,k} \cdot s_k^2 + C_{Y3,k} \cdot s_k^3 \\ C_{Z0} + C_{Y1,k} \cdot s_k + C_{Y2,k} \cdot s_k^2 + C_{Y3,k} \cdot s_k^3 \end{bmatrix} + v_k \\ &= \begin{bmatrix} s_k & s_k^2 & s_k^3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & s_k & s_k^2 & s_k^3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & s_k & s_k^2 & s_k^3 \end{bmatrix} \cdot x_k + \begin{bmatrix} C_{X0} \\ C_{Y0} \\ C_{Z0} \end{bmatrix} + v_k, \end{aligned} \tag{2.10}$$

where k denotes the iteration step of the Kalman filter, s_k is a system input denoting the parameter value of the polynomial curve, and w_k and v_k are the prediction and measurement noises with covariance Q_k and R_k, respectively. Under ideal conditions where the prediction noise is ignored, the system state does not change during the prediction step. This is reasonable since the coefficients of the polynomial curve are

only affected by the data points that are observed. The coefficients should be adjusted only when a new data point has been added.

Measurement matrix H_k is a function of the time-varying system input s_k. Ideally, s_k is an arc-length parameter of the spline curve that corresponds to the measurement point y_k. However, since it is impossible to calculate the arc-length of the curve before finishing the approximation process, I approximate the curve by using arbitrary, strictly-increasing parameters first, and then I re-parameterize the curve by the arc-length after finishing the approximation process. To simplify the re-parameterization process, I use the chord-length from the initial point as the parameter value, s_k, which is an approximated value of the arc-length.

Observability Proof

The suitability of the proposed system model can be proven by checking the observability of the system. Observability is a necessary condition for the Kalman filter to work correctly, and we say that a linear system is observable if there is a finite number of steps n so that knowledge about the input sequence $u_0, ..., u_{n-1}$ and the output sequence $y_0, ..., y_{n-1}$ is sufficient to determine the initial state of the system, x_0. Therefore, the system is observable if x_0 is uniquely determined by

$$\begin{bmatrix} y_0 \\ \vdots \\ y_{n-1} \end{bmatrix} = O_n x_0 + T_n \begin{bmatrix} u_0 \\ \vdots \\ u_{n-1} \end{bmatrix}, \tag{2.11}$$

where O_n is a matrix that maps the initial state x_0 into the resulting output over $[0, n-1]$ and T_n is a matrix that maps the input to the output over $[0, n-1]$. For

our system, the above problem can be represented as

$$\begin{bmatrix} y_0 \\ \vdots \\ y_{n-1} \end{bmatrix} = \begin{bmatrix} H_0 \\ \vdots \\ H_{n-1} \end{bmatrix} x_0 + \begin{bmatrix} C_0 \\ \vdots \\ C_0 \end{bmatrix}, \tag{2.12}$$

where $C_0 = [C_{X0}, C_{Y0}, C_{Z0}]^{\mathrm{T}}$. For $n = 3$, the initial state $x(0)$ is uniquely determined if and only if $N(O_3) = 0$ and, equivalently, $\mathrm{Rank}(O_3) = 9$. The matrix O_3 is composed of three observation matrices: H_0, H_1 and H_2. It can be easily proved that $\mathrm{Rank}(O_n) = 9$. By definition, since $s_k > s_{k-1}$ for every k, every row vector for matrix O_n is independent relative to the other row vectors. Therefore, I can prove that the proposed system is observable.

Error Covariance

In order to obtain good results from the Kalman filter, the proper error covariance matrices, Q_k and R_k, need to be set up. First, the prediction error should be zero (and thus $Q_k = \mathbf{0}$) for this problem since the statistics of the state values do not vary during the prediction step.

The measurement error covariance matrix, R_k, is determined according to the statistics of the vehicle pose data and the MLS data. For example,

$$R_k = \begin{bmatrix} 0.05^2 & 0 & 0 \\ 0 & 0.05^2 & 0 \\ 0 & 0 & 0.05^2 \end{bmatrix}. \tag{2.13}$$

Processing Outlier Points

The Kalman filter has the inherent ability to handle outliers by allocating small gains to measurements that are far from the predicted system state. However, outliers that

are too large or too frequent may force the Kalman filter to diverge. To address this problem, outliers are detected and isolated so that they are not used for state update, by applying a normalized innovation squared (NIS) test to every measurement point before updating the system state using the measurement point. The NIS test is defined as follows:

$$\text{NIS}_k = (y_k - H_k\hat{x}_{k|k-1})^T S_k^{-1}(y_k - H_k\hat{x}_{k|k-1}) > \lambda, \tag{2.14}$$

where

$$S_k = H_k P_{k|k-1} H_k^T + R_k. \tag{2.15}$$

The measurement point that is larger than a threshold λ is considered to be an outlier, and it is discarded. The threshold value λ should be carefully determined since a value that is too large for λ negatively impacts the ability to detect for an outlier, while the Kalman filter may diverge with a too small value for λ due to frequent removal of normal measurement points.

2.4.3 Curve Transition

Based on the proposed system and measurement model, covariance matrices, and outlier point process algorithm, the KF approximates the road geometry point data that are given one by one to a polynomial curve. At a certain point, the KF should stop the current approximation and should start to generate a new polynomial curve since one cubic polynomial curve cannot express all points. The condition and procedure for the curve transition are discussed below.

At the end of every step of the KF, two kinds of maximum distance between the curve and the data points are calculated in order to decide whether the new curve

approximation should be started, i.e.,

$$\begin{aligned} D_{\max,XY} &= \max_i \left\| L_{XY}(\bar{s}_i) - G_{XY,i} \right\|, \; i = 1, ..., k, \\ D_{\max,Z} &= \max_i \left\| L_Z(\bar{s}_i) - G_{Z,i} \right\|, \; i = 1, ...k, \end{aligned} \tag{2.16}$$

where

$$L_{XY}(s) = \begin{bmatrix} C_{X0} + C_{X1} \cdot s + C_{X2} \cdot s^2 + C_{X3} \cdot s^3 \\ C_{Y0} + C_{Y1} \cdot s + C_{Y2} \cdot s^2 + C_{Y3} \cdot s^3 \end{bmatrix} \tag{2.17}$$

and

$$L_Z(s) = C_{Z0} + C_{Z1} \cdot s + C_{Z2} \cdot s^2 + C_{Z3} \cdot s^3 \tag{2.18}$$

are the polynomial curve on the XY- and Z-planes, respectively, while $G_{XY,i}$ and $G_{Z,i}$ refer to the i-th data point on the XY- and Z-plane, respectively. Therefore, $D_{\max,XY}$ and $D_{\max,Z}$ refer to the maximum distance errors between the curve and the data points on the XY- and Z-plane, respectively. The reason to calculate the distance error on the XY- and Z-plane separately is that the accuracy requirements for the XY- and Z-plane can be different. In general, autonomous vehicle systems require accuracy at the centimeter-level in the XY-plane since the map accuracy in the XY-plane affects vehicle safety in applications that involve vehicle localization or vehicle motion planning. On the other hand, a relatively low accuracy is required for the Z-plane since generally the height information of the road is generally used in non-safety-related applications such as for fuel management systems.

Ideally, the curve transition should be conducted at the moment when at least one of the two distance error exceeds a pre-defined threshold, i.e.,

$$D_{\max,XY} > \chi_{XY} \text{ or } D_{\max,Z} > \chi_Z, \tag{2.19}$$

However, the maximum distance error can occasionally exceed the threshold even though all data points belong to one polynomial curve due to stochastic errors that

are included in the data. In this case, the errors have to be ignored, and the KF should be continued. For this reason, a heuristic test is applied for the curve transition as follows:

i. A counter is initialized to zero before starting the KF.

ii. At the end of the every step of the KF, the maximum distance errors, $D_{\max,XY}$ and $D_{\max,Z}$, are calculated.

iii. The counter increases if $D_{\max,XY} > \chi_{XY}$ or $D_{\max,\mathrm{Z}} > \chi_Z$ and it is set to zero otherwise.

iv. If the counter exceeds a threshold N_{buff}, the curve transition is triggered.

In summary, the curve transition is triggered when the maximum distance errors exceed the threshold N_{err} times in a row. Once the curve transition is triggered at step k, the curve approximated at step $k-N_{\text{buff}}$ is saved since the curve approximated from step $k-N_{\text{buff}}$ to k is inaccurate. Therefore, state vectors and covariance matrices for the KF for recent N_{buff} steps should be stored in the buffer, and after the current polynomial curve is saved, the KF is initialized and a new curve approximation process begins.

In addition to the normal case, there is another case where the curve transition is needed. In a road, the number of lanes can vary, as shown in Fig. 2.7. In this case, the curve approximation process has to be stopped at the point where the road line disappears, and a new curve has to be started from the point where the road line reappears. In order to handle this case, at every step of the approximation process, the distance between the current data point and the next point is calculated. If the distance is larger than a threshold, i.e.,

$$\|G_{i+1} - G_i\| > \eta, \tag{2.20}$$

which means that the road line disappears at the current point, the curve transition is triggered. Since the new curve is discontinuous with the previous curve, the curve parameter representing the arc-length is set to zero at the start point of the new curve.

2.4.4 Arc length parameterization

The remaining portion of the procedure for road modeling involves arc-length parameterization. As mentioned in the previous sections, the curve generated from the sequential approximation algorithm is parameterized according to the chord length as an approximation of the arc-length. However, since accurate arc-length information is useful for many intelligent vehicle applications, I parameterize the curve in terms of the arc-length as the final task for the road modeling. Many studies have investigated the arc-length parameterization task for a curve, and in this dissertation, I use the algorithm proposed in [47] since it can process a large quantity of map data in a simple and efficient manner. The arc-length parameterization of a curve can be constructed by adhering to the following two-step process:

i. Compute the arc length s as a function of parameter $t : s = A(t)$. Since s is a strictly increasing function of t, there is an one-to-one correspondence between s and t.

ii. $t = A^{-1}(s)$, inverse of the arc-length function. This function is well defined and monotonically increases for cubic splines. By substituting $t = A^{-1}(s)$ into $Q(t)$, I obtain a curve that is parameterized by the arc length s, $P(s) = (X(A^{-1}(s), Y(A^{-1}(s), Z(A^{-1}(s)))))$, where $s \in [0, D]$ and D is the total length of the curve.

Figure 2.6 Probe vehicle equipped with a 3D Lidar and a GPS+INS positioning system.

Further details of the algorithm are provided in the literature [47]. After conducting the arc length parametrization, we obtain the final arc length-parameterized cubic spline curve for the data points of the road geometry.

2.5 Experimental Validation

In this section, I provide the results of the experiment that was carried out to evaluate the performance of the proposed road map generation system. First, I introduce the probe vehicle and sensor configuration used in the experiments. I then present the results of 3D Lidar-based data acquisition and data processing. I also evaluate the accuracy, storage efficiency and usability performance of the road modeling system. Finally, I provide an autonomous driving test based on the proposed road map to validate the practicality of the proposed road map system.

2.5.1 Experimental Setup

The raw data required to evaluate the proposed road map generation system was collected using a probe vehicle equipped with a 3D Lidar (Velodyne HDL-64E) and

Table 2.2 Specification of the 3D Lidar.

Horizontal field of view	Vertical field of view	Distance accuracy	Measurement rate
360 deg.	26.8 deg.	<2 cm	>1.3 M points/sec.

Table 2.3 Specification of the vehicle positioning system.

Measurement	Frequency	Accuracy (RMS)
Position	100 Hz	2 cm
Yaw	100 Hz	0.1 deg.
Roll/pitch	100 Hz	0.03 deg.

a GPS+INS vehicle positioning system (OXTS RT3002) as shown in Fig. 2.6. The Velodyne HDL-64E is a 64-layer Lidar, and it was configured to rotate 360 degrees to emit 64-layer laser beams with a minimum of 1.3 million points measured per second. OXTS RT3002 is a unified GPS+INS system that provides a highly accurate 6D vehicle poses by combining the GPS and INS data. The specifications of these two measurement system are summarized in Tables 2.2 and 2.3, respectively. In this dissertation, raw data was collected with a 10 Hz rate, and the prove vehicle was driven at a normal driving speed of 30 km/h to 80 km/h depending on the the road type and conditions. All of the experiments were carried out on a PC with a 3.40 GHz i7-4770 CPU. The parameters χ_{XY} and χ_Z in Eqs. (2.19) were set to 0.1 m and 0.3 m respectively, and η in Eqs. (2.20) was set to 10m.

2.5.2 Data Acquisition and Processing

Two different data sets were collected from the Incheon International Airport Express (IIAE) and Seoul National University (SNU) ring road, respectively. The IIAE is

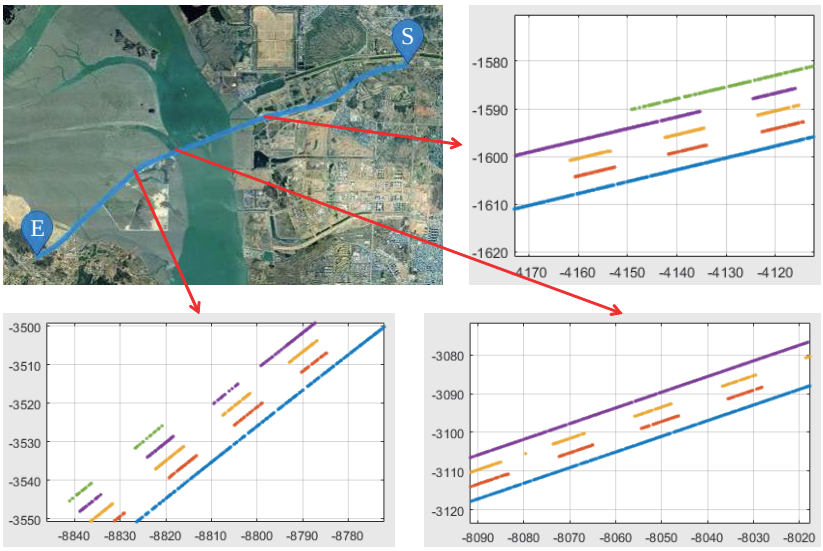
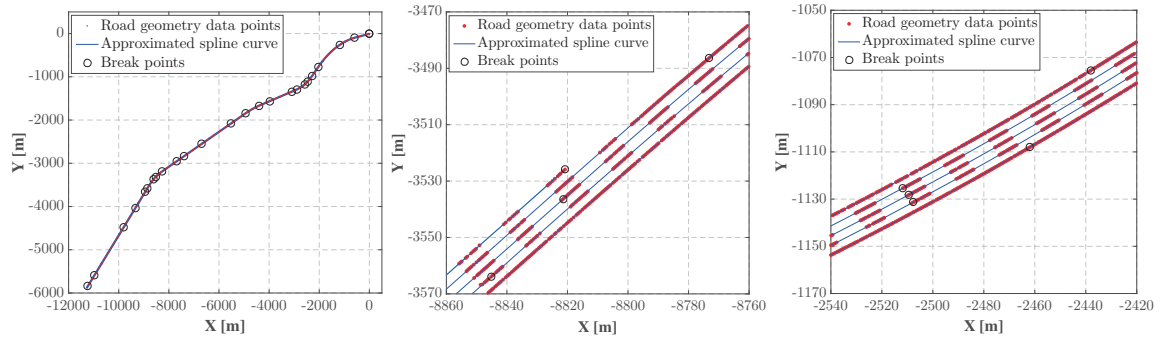


Figure 2.7 Data processing result for the IIAE data set.

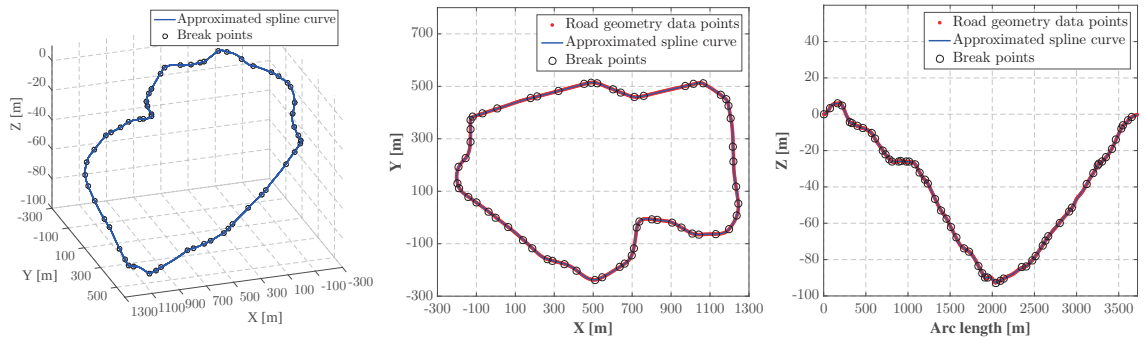
a *type I* road that has four lanes, and thus is suitable to evaluate the road map generation system including the MLS-based data acquisition system. The IIAE data set covers a road length of approximately 13 km and contains about 5,900 frames of synchronized 3D MLS (7.7 billion points) and 6D vehicle pose data. Fig. 2.7 shows a qualitative result of the data processing algorithm when applied to the IIAE data set. From total 7.7 billion laser points, about 65,000 were extracted after executing the data processing algorithm. It is impossible to present the quantitative accuracy of the data acquisition and processing result, since there is no way to measure more accurate coordinates of the lane markings than the RTK-GPS + 3D Lidar configuration that was used in this dissertation. The data processing operates in off-line mode. In the experiment, it took about 38 seconds to process 1 km road of the IIAE data set, where the algorithm was implemented based on the point cloud library [48].

The SNU ring road is a *type II* road and is more appropriate to evaluate the road modeling system since it contains various curves and height variations. The SNU data



(a) (b) (c)

Figure 2.8 The result of the road modeling for the IIAE data set. (a) Overall map, and (b) and (c) enlarged maps.



(a) (b) (c)

Figure 2.9 The result of the road modeling for the SNU data set. (a) 3D road map, (b) road map projected on XY plane, and (c) height of the road.

set consists of a road with a distance of approximately 3.7 km and contains about 6,900 vehicle pose data in intervals of about 0.5 m.

2.5.3 Road Modeling

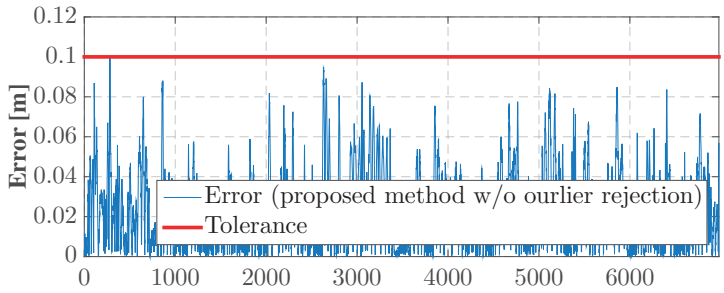
Both the IIAE and SNU data sets were modeled using spline curves by using the proposed road modeling algorithm. Figs. 2.8 and Fig. 2.9 show the results of road

modeling where the XY-plane and Z-axis accuracy thresholds, $D_{\mathrm{max,XY}}$ and $D_{\mathrm{max,Z}}$, were set to 0.1 m and 0.3 m, respectively. Since the IIAE data set contains four lanes, a total of five spline curves were generated, whereas only one spline curve was generated for the SNU data set. The dots in the figures denote the point data, and the lines denote the approximated spline curve. The circles indicate the break points of the spline curves. The road modeling process operates in off-line mode. In the experiment, it took about seven seconds to process 1 km road of the IIAE data set.

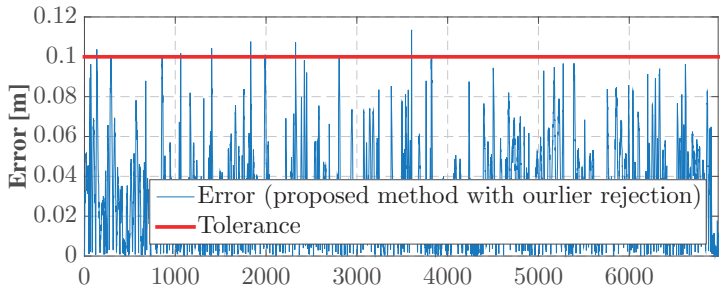
To quantitatively evaluate the performance of the proposed road modeling system in terms of the accuracy, storage efficiency and usability, I conducted a comparison with a B-spline-based state-of-the-art algorithm that was proposed in [14]. The B-spline-based algorithm had been proven to outperform various previous algorithms, including algorithms using polygons, natural cubic splines, or cubic B-splines with a constant interval.

Accuracy

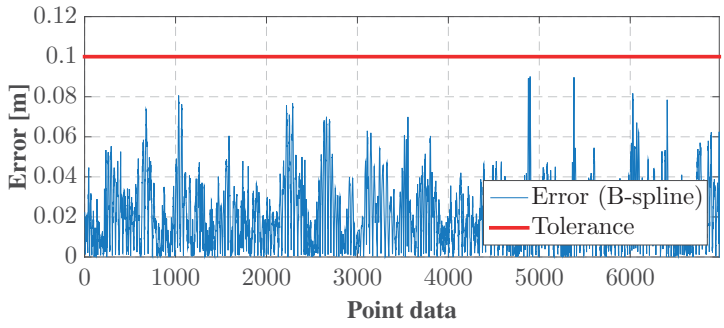
In terms of the accuracy, a major goal of the road modeling system is to generate a curve model for the distance errors between the road geometry data points and the curve not to exceed a preset tolerance, e.g., 0.1 m. Fig. 2.10 shows the modeling accuracy of the proposed algorithm and the B-spline-based algorithm when the SNU data set is used with 0.1m XY-plane accuracy tolerance. The horizontal axis of the figure refers to each data point and the vertical axis refers to the distance error between the data points and the generated curve. Note that two different results are presented for the proposed algorithm depending on whether or not the outlier rejection scheme was used. In the case where the outlier rejection was used, six points were found to violate the 0.1m tolerance. However, exceeding the tolerance does not



(a)



(b)



(c)

Figure 2.10 Errors between road geometry data points and road models generated by (a) proposed algorithm without outlier rejection, (b) proposed algorithm with outlier rejection, and (c) B-spline-based algorithm. Both methods satisfies the 0.1 m accuracy constraint.

necessarily mean that the accuracy is degraded but rather that the outliers were detected and rejected by the algorithm. I found that all points that violated the

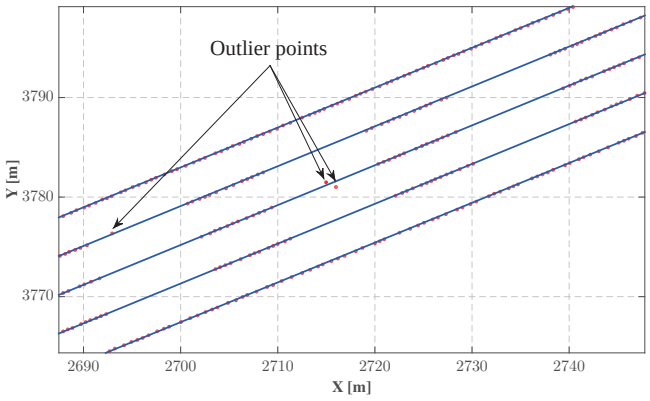
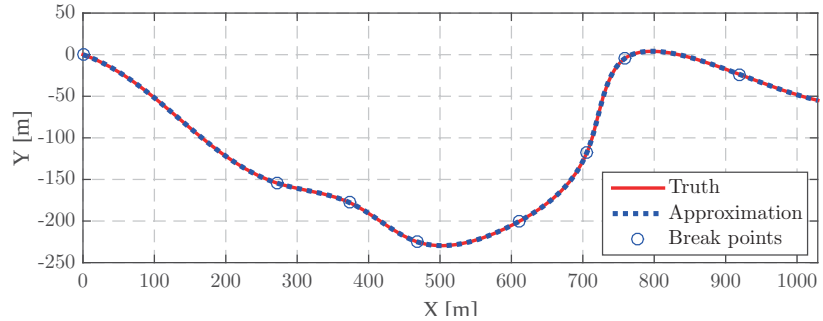


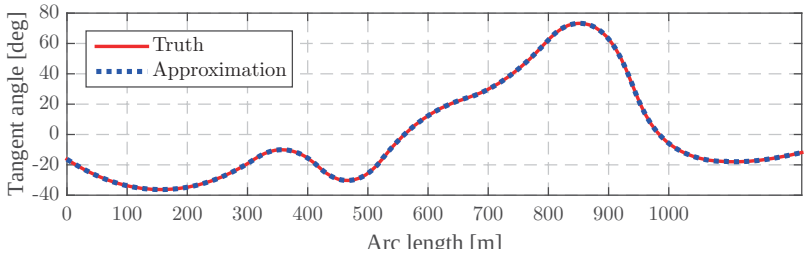
Figure 2.11 Example of the outlier rejection which shows that the outliers were effectively detected and rejected.

tolerance coincide with outliers that had been detected during the road modeling process, as the example in Fig. 2.11 shows. The outlier rejection affects the storage efficiency of the map, and in this example, the entire road was modeled with 55 piecewise polynomials in the case where outliers were rejected while a total of 67 piecewise polynomials were used in the case where outliers were not rejected. As a consequence, we can see that the proposed road modeling algorithm successfully detects and rejects outliers that are contained in data points and also accurately and efficiently represents data points.

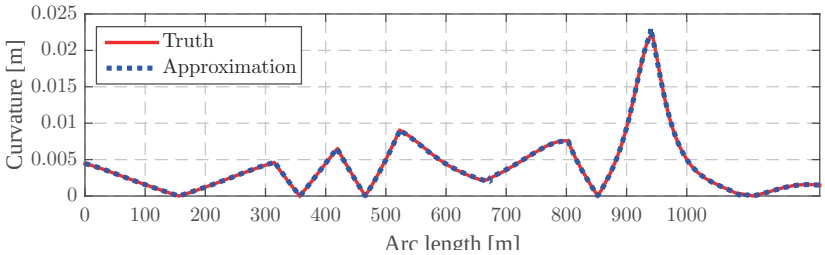
Accurate coordinates as well as accurate tangent angle and curvature information is very important for various intelligent vehicle applications. Such information is essential for safety-related systems, including motion planning systems. Adhering to the tolerance requirements does not necessarily ensure the accuracy of the tangent angle and curvature, and therefore, I also evaluated the road modeling algorithm in terms of the accuracy of the tangent angle and curvature. It is difficult to evaluate the accuracy of the tangent angle and curvature by using real road geometry data points



(a)



(b)



(c)

Figure 2.12 Qualitative result of road modeling in terms of (a) coordinates, (b) tangent angle, and (c) curvature.

since it is impossible to know the exact tangent angle and curvature values of the real road. Thus, I generated artificial road geometry data points for which I know the exact tangent angle and curvature values by first generating an artificial spline curve, shown in Fig. 2.12(a), and then I sampled the curve at every 1 m. To ensure that the data is close to the real values, Gaussian random noise with a standard deviation of

Table 2.4 Accuracy of tangent angle and curvature of the proposed road model and the B-spline-based road model.

		Mean	Std.	RMS	Max
Tangent angle	Proposed	0.898E-02	1.88E-02	1.59E-02	2.67E-01
error [deg.]	B-spline	3.54E-02	3.86E-02	5.24E-02	3.13E-01
Curvature	Proposed	1.67E-05	0.682E-04	0.496E-04	0.841E-03
error [m^{-1}]	B-spline	7.98E-05	1.23E-04	1.47E-04	1.24E-03

0.05m was added to all points. The artificial road geometry data was approximated with a set of polynomials that make use of the proposed road modeling algorithm, and the geometry information that is extracted from the curve was compared to the true value of the road geometry. Fig. 2.12 presents the results, and we can see that the road model accurately expresses all the true road geometries.

A quantitative evaluation was also carried out. Table 2.4 shows the accuracy of the tangent angle and curvature of the road models that were generated using the proposed algorithm and B-spline-based algorithm, where the tolerance of the algorithm was set to 0.1 m. The results show that the proposed algorithm generated a more accurate road model than the B-spline-based algorithm. Furthermore, the levels of maximum error in the tangent angle and curvature were 10^{-1} and 10^{-3} respectively. In general, such levels are acceptable for the errors in intelligent and autonomous vehicle applications.

Storage Efficiency and Usability

The storage efficiency and usability were evaluated together. First, the SNU data set was modeled using the proposed algorithm and the B-spline-based algorithm to com-

Table 2.5 Storage efficiency and usability of the proposed road model and the B-spline-based road model.

	# of segments	# of floating-point numbers	Computational time (μs)			
			Position	Tangent angle	Curvature	Closest point
Proposed	55	715	0.68	0.47	0.59	16
B-spline	178	728	32	82	146	630
B-spline (precomputed derivatives)	178	1805	32	29	52	280

pare the storage efficiency and usability of the two models. In this dissertation, the storage efficiency is measured by the number of floating-point numbers that are used to express the road model. In the case with the proposed set of piecewise polynomial forms, a piecewise polynomial requires thirteen floating-point numbers (one beginning parameter and twelve polynomial coefficients). Therefore, the set of piecewise polynomial forms requires totally $13\times$[*number of piecewise polynomials*] floating-point numbers. The number of required floating-point numbers for the B-spline form depends on the number of knots and control points. Since each control point contains three floating point numbers (X,Y and Z coordinates), the B-spline requires a total of [number of knots]$+3\times$[*number of control points*] floating-point numbers.

The usability of such a system for intelligent and autonomous vehicles is inversely proportional to the computational time required to calculate the essential information from the road models. Four essential pieces of information were selected: position, tangent angle and curvature at an arbitrary point on a curve, and closest point on a curve from an external point. The position, tangent angle and curvature information can be analytically calculated using both the proposed and the B-spline curves. However, the B-spline curve requires complex recursive calculations to obtain this information [40], while the proposed method only requires simple calculations since the first and second derivative of the curve are easily calculated from the polynomials.

The closet point on a curve should be calculated using a numerical method for both curves. In this dissertation, I used Newton's method to evaluate both curves on equal terms (with 10^{-6} tolerance), and the experiments were carried out on a PC with a 3.40 GHz i7-4770 CPU.

Table 2.5 shows the results of the experiment. Note that the B-spline was evaluated using two different modes. In the normal mode, only the knots and control points of the B-spline curve itself were stored in the memory. On the other hand, in the second mode, the knots and control points for the first and second derivatives of the B-spline curve were precomputed and additionally stored in the memory. The precomputation of the derivatives increases the storage space that is required, however, it decreases the computational time. The results show that the proposed algorithm outperforms the B-spline-based algorithm in both modes in terms of both the storage efficiency and usability. In particular, the proposed road model dramatically decreases the computational time needed to calculate the various pieces of information for intelligent and autonomous vehicles. In terms of storage efficiency, note that the B-spline-based algorithm requires a total of 178 curve segments, whereas the proposed algorithm requires only 55 curve segments. This means that, in the case in which the curve is first approximated as a B-spline curve and converted to the piecewise-polynomial form, a large amount of storage is required than when using the proposed algorithm.

2.6 Summary

This chapter presented a road map generation system that simultaneously considers the accuracy, storage efficiency and usability to generate high-precision lane-level road maps for use in intelligent vehicle systems. The overall system is composed of three subsystems, including data acquisition, data processing and road modeling systems.

The MLS data-based data acquisition and processing system efficiently acquires accurate road geometry data by integrating 6D vehicle pose and 3D MLS data. The main contribution of this chapter is to present a road modeling algorithm that models road geometry data as a form consisting of sets of piecewise polynomial curves that are more appropriate for use in intelligent vehicle systems than those of previous road models, such as clothoids or B-splines in terms of usability. The proposed road modeling algorithm maximizes the storage efficiency of the map by minimizing the number of piecewise polynomials needed to express the map. The experiments were conducted to demonstrate that the map that is generated by the proposed algorithm is more effective than those obtained using previous algorithms in terms of the storage efficiency and usability.

Chapter 3

Accurate and Cost-Effective Vehicle Localization in Featureless Environments

Vehicle localization is a key component of the automated driving system. Many automated driving functions such as path planning and motion planning benefit from accurate vehicle position information. Conventionally, integration of GPS and DR has been widely used for vehicle localization [49–52]. However, GPS/DR systems are not appropriate for automated driving, since the GPS/DR systems cannot provide accurate and reliable position information during long time GPS outage.

In order to provide more accurate and reliable position information, localization systems utilizing a map and various perception sensors such as Lidar and camera have been actively studied. In this approach, a precise road map is generated beforehand, and the localization is performed on the map by matching the real-time perception sensor information and the information stored in the map. In terms of the accuracy, the 3D Lidar is one of the most promising sensors for the mapping and localization

problem. The 3D Lidar provides accurate, reliable and plenty 3D range information about surrounding environment, and the performance is not affected by external factors such as intensity of illumination and weather. However, the 3D Lidar is not appropriate to use for commercial vehicles, because of its high cost.

In order for accurate, but cost-effective vehicle localization, in recent years, vision-based vehicle localization has been actively studied [53–58]. The vision-based localization systems use various image features such as lane markings, road markings, and low-level image features for the mapping and localization. More specifically, these feature information are stored in the map, and the feature information extracted from camera images are matched to the stored features to obtain accurate vehicle position. Therefore, the performance of the localization mainly depends on how many features exist on road and in surrounding of the road, and how the features are different for different road sections. Under the environments with rich feature information such as urban environment, there have been many studies on vision-based vehicle localization. However, relatively fewer studies have been conducted for the environment where feature information for the localization is poor, which is named as a featureless environment in this paper. For example, in a highway environment, in many road sections there exist no other feature information except the lane markings.

This paper proposes a precise vision-based/map-aided vehicle localization system for the featureless environment. Data from four different types of sources are integrated for this goal: a low-cost GPS, in-vehicle network sensors, a monocular camera, and a high-precision lane-level road map. Even though this low-cost sensor configuration is desired for automated vehicle systems, it causes a so-called measurement ambiguity problem. We show that a conventional particle filter (PF) cannot solve the ambiguity problem, and propose a particle filter with modified likelihood

function based on a probabilistic particle association to solve the problem. Through experiments using real road data, it is demonstrated that the proposed localization system provides accurate vehicle pose information for automated driving under the featureless environment.

The remainder of this paper is organized as follows. Section 3.1 presents related works. In section 3.2, an overview of the localization system is presented. In section 3.3, overall process of the proposed PF-based localization algorithm is described. In section 3.4, detailed map-image matching process and probabilistic measurement association scheme is presented. The accuracy of the proposed localization system is evaluated in section 3.5 through the experiments, and section 3.6 summarizes this paper.

3.1 Related Works

There have been lots of studies for accurate vehicle localization as shown in Fig. 3.1. Integration of GPS and DR (dead reckoning) has been widely used for vehicle localization [49–52, 59]. The GPS provides global positioning information, however, the update rate is low and the error can be increased significantly by GPS signal outage and multi-path effect for short period of time. The DR estimates the vehicle pose using vehicle motion sensors with high frequency. However, stand-alone DR suffers from long-term error accumulation problem. These complementary characteristics of the GPS and DR make it possible to enhance the localization performance by the integration of the GPS and DR. However, during long time GPS outage, the GPS/DR system cannot correct the diverging DR error. Fig. 3.2 show an example of the visual odometry-based DR. For a highway road, a dense visual odometry algorithm proposed in [60] was applied using monocular camera images with 30 Hz frame rate. The

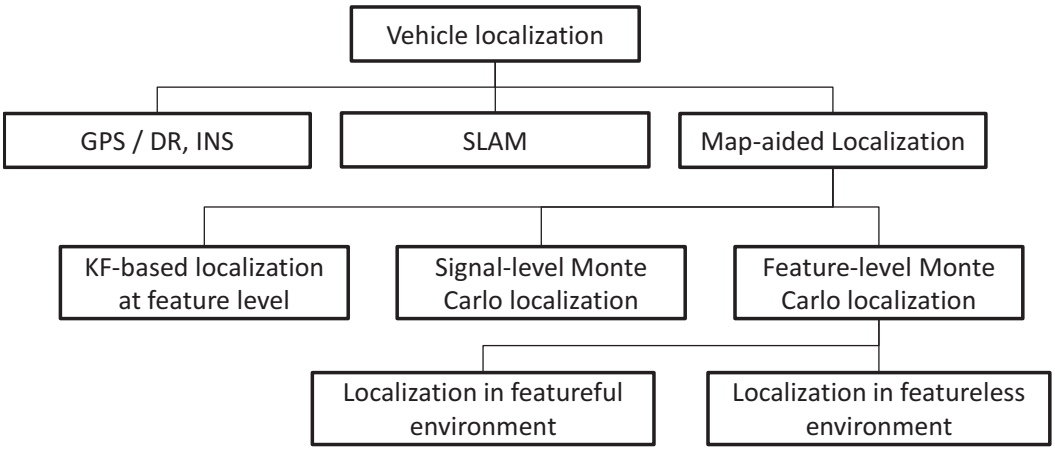
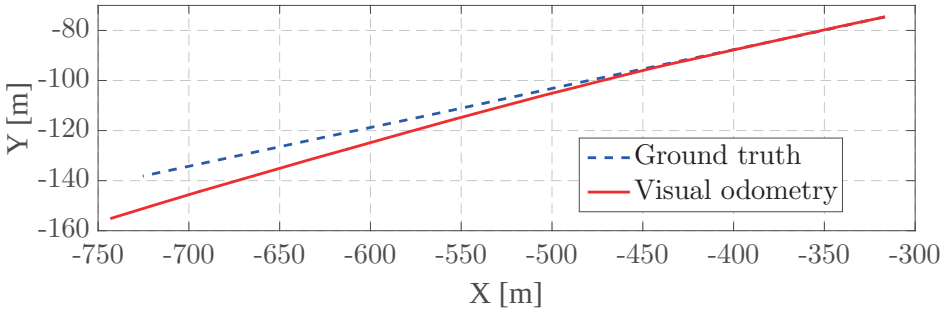


Figure 3.1 Categorization of localization researches.

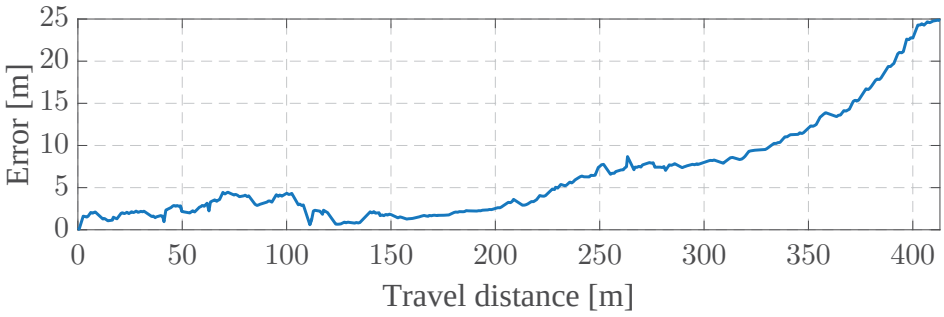
Fig. 3.2a presents the ground truth position of the vehicle and the result of the visual odometry algorithm, and Fig. 3.2b presents the error between the ground truth and the visual odometry result. The result shows that the localization error drastically increases during long time GPS outage.

Simultaneous localization and mapping (SLAM) is a solution for accurate vehicle localization without a GPS. The SLAM generates a map in real-time using various sensors such as laser scanners and camera, and simultaneously localizes the vehicle in the map [61–63]. The SLAM can effectively provide accurate localization result for unknown environment, however, the localization process may get easier if a precise map is given beforehand.

In the map-aided localization approach, a precise and accurate map is given, and the localization is conducted based on the map by utilizing various perception sensors such as laser scanners and cameras. This approach can be categorized into three groups: 1) Kalman filter-based localization at the feature level, 2) signal level Monte Carlo localization, and 3) feature-level Monte-Carlo localization.



(a)



(b)

Figure 3.2 Performance of stand-alone visual odometry.

In the Kalman filter-based localization at the feature level, various features are extracted from various perception sensor measurements, and they are associated with land marks contained in the map. Using the association result, the vehicle position state is updated based on a Kalman filter or its branches [53, 54, 64, 65]. This method has an advantage that the vehicle position can be updated by simple Kalman filter calculations. However, because of the inherent limitation of the Kalman filter, it is not appropriate to apply to nonlinear or non-Gaussian systems.

Because of the limitations of the Kalman filter-based methods, there have been many studies on Monte Carlo localization. In this approach, the nonlinear or non-Gaussian system and measurement models are approximated by a Monte Carlo sampling manner. The Monte carlo localization has two different modes: signal-level local-

ization, and feature-level localization. In the signal-level Monte Carlo localization, the raw measurements of the perception sensors are used for so-called perception-to-map association [66–68]. Since the entire raw measurements are used for data association, complex and inconvenient algorithms for feature extraction are not needed. However, it is inefficient in terms of computational complexity, since the entire raw measurements have to be processed at once.

In order to compensate the disadvantage of the signal-level Monte Carlo localization, recently, feature-level Monte Carlo localization has been actively studied. In this approach, various feature information extracted from the perception sensors are used for perception-to-map association. Various perception sensors including camera, radar, and LIDAR are used to build a precise digital map and for ego-vehicle localization on the map. Among these sensors, camera is most widely used since a camera provides abundant information with low-cost compared to the other sensors. The vision-based localization systems use various image features such as lane markings, road markings, and low-level features for the mapping and localization. Under the environments with rich feature information such as urban environment, there have been many studies on vision-based precise vehicle localization [53–57, 67, 68]. However, relatively fewer studies have been conducted for the environments where feature information for the localization is poor, which named a featureless environment in this chapter. For example, in a highway environment, in many road sections there exist no other feature information except the lane markings. This chapter presents a vehicle localization system for the featureless environments.

3.2 System Overview

3.2.1 Test Vehicle and Sensor Configuration

A test vehicle was configured based on various low-cost sensors for the vehicle localization. First, I utilized some vehicle motion sensor information that the electronic stability control (ESC) module of the vehicle contains. These sensor information can be easily captured from the in-vehicle network (controller area network, CAN). Two types of sensor information (wheel speed sensors and a yaw rate sensor) were used for dead reckoning of the vehicle, and the specifications of the motion sensors are listed in Table 3.1.

A low-cost GPS was used for basic positioning information. Specifically, the GPS positioning information was used to initialize the localization system, and to prevent the localization error from diverging. I installed JYSYSTEM B20, and the specifications of the GPS are given in Table 3.2. The B20 support the DGPS mode, however, the DGPS mode was turned off in this configuration.

The test vehicle equipped with a monocular camera. The camera provides images of a 90 degree field of view (FOV) in the front of the vehicle. Multiple cameras can be used, however, I found that single camera provides enough localization accuracy and it is favorable to minimize the computational burden. The detailed specifications of the camera are listed in Table 3.3.

3.2.2 Augmented Road Map Data

The road map presented in the previous chapter represents the road lines as a set of piecewise polynomials. In other words, the road map only contains basic road geometry information such as position coordinates, tangent angle, and curvature. However, this basic road geometry information is insufficient to utilize for precise

Table 3.1 Specification of in-vehicle network sensors.

Sensor	Range	Frequency	Resolution	Noise RMS
Wheel speed sensor	0∼130	50 Hz	0.035 m/s	0.3 m/s
Yaw rate sensor	120	50 Hz	0.0625 deg/s	0.5 deg/s

Table 3.2 Specification of low-cost GPS.

Position accuracy	2.5m circular error probability (CEP)
Update rate	1∼10 Hz

Table 3.3 Specification of monocular camera.

Field of view	Frame rate	Resolution
90°	30 Hz	1280×1024

vehicle localization. Therefore, the road map is augmented so that it contains more information that is useful for precise vehicle localization. Two types of information are additionally included. First one is dashed lane marking information. A large portion of real roads contains dashed lane markings. Therefore, the dashed lane marking information can increase both the lateral and longitudinal vehicle localization performance in many road environments. Second is road markings information. Various road markings such as arrows and texts on the roads can be used as landmarks for the localization. Especially, the road marking information is useful for increasing the longitudinal localization performance in solid line road sections.

In order to include the dashed lane marking information in the road map, I first found the points corresponding to the beginning and end point of the lane marking

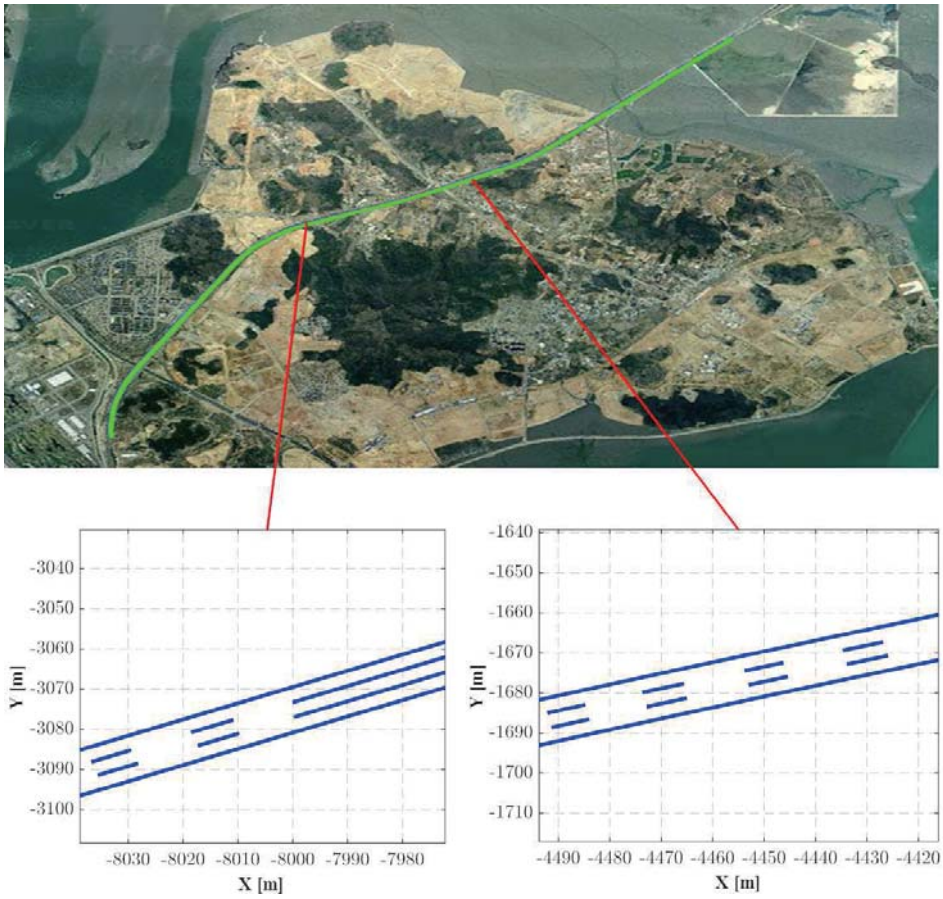


Figure 3.3 Road map including dashed lane marking information.

using the road geometry point data obtained in chapter 2. Then a set of pair of points denoting the beginning and end points of the lane markings is obtained for a road line, i.e.,

$$LMp^i = \left\{ \{(x_b, y_b), (x_e, y_e)\}_j^i, j = 1, ..., N_i \right\}, i = 1, ..., N_{\text{lane}}, \tag{3.1}$$

where i denote the index of the road lines and j means the index of the lane markings. Next, in order to apply this information to the polynomial-based road map, the curve parameters (arc-length parameters) are founded and the sets of parameters

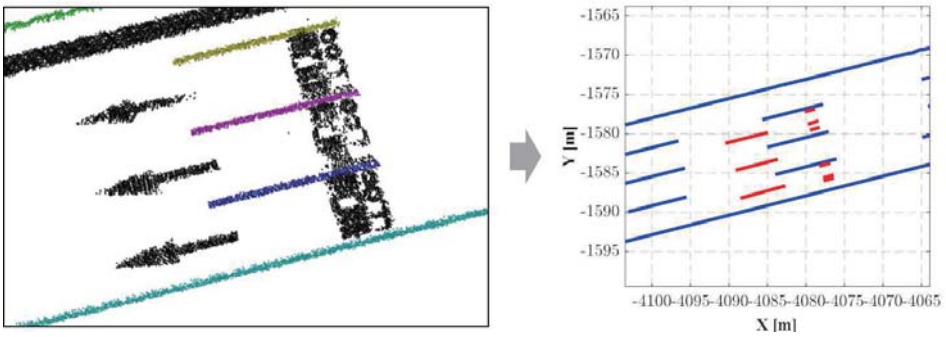


Figure 3.4 Road map including road marking information.

representing the beginning and end point of the lane markings are generated as below:

$$LMpam^i \left\{ \{s_b, s_e\}_j^i , j = 1, ..., N_i \right\}, i = 1, ..., N_{\text{lane}}, \tag{3.2}$$

where $\{s_b, s_e\}_j^i$ denote the curve parameters corresponding to the beginning and end point of the lane markings. From this approach, the dashed lane marking information can be easily extracted from the road map as shown in Fig. 3.3.

The road marking information is extracted manually from the LIDAR data as shown in Fig. 3.4. As will be explained later, the road map data should be transformed to an image plane for map to image matching. Therefore, to reduce the computational burden required for the transformation process, only the vertical line components of the road markings are extracted from the LIDAR data. Extracted road markings are stored in the road map by assigning the index number of the corresponding curve segment to each road marking. Finally, the road marking information can be easily extracted from the map as shown in Fig. 3.4.

Including the additional information in the map may increase the data size. However, the dashed lane markings are represented as compact forms as explained above. For the Incheon international airport expressway, I examined the required data to build the map. For the original map that does not contain the dashed lane marking

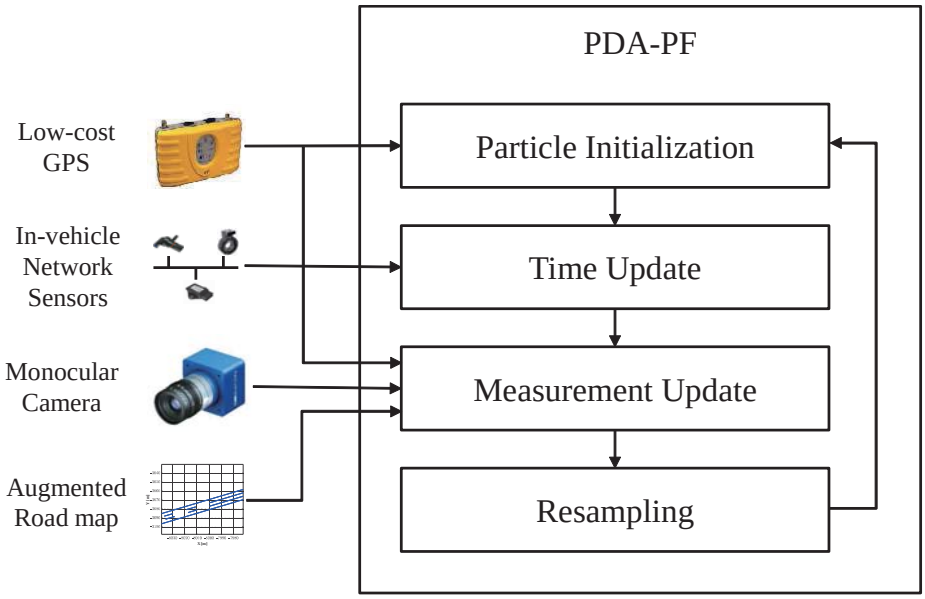


Figure 3.5 Overall localization system architecture.

information, average 700 bytes data per 1 km road was required for a lane. For the augmented map, average 1.4 Kbytes data per 1 km road was additionally required for a lane.

3.2.3 Vehicle Localization System Architecture

The overall architecture of the proposed localization system is described in Fig. 3.5. There are four types of data sources for the vehicle localization: low-cost GPS, in-vehicle network sensors, monocular camera, and augmented road map. The four types of data are integrated by a particle filter to estimate accurate 3D vehicle pose (position, heading). The low-cost GPS is used to initialize particles of the filter and to prevent the localization error from diverging at the measurement update step. The in-vehicle network sensor information (wheel speed, yaw rate) is used to capture the vehicle motion within a short period of time. Specifically, the motion sensors are used

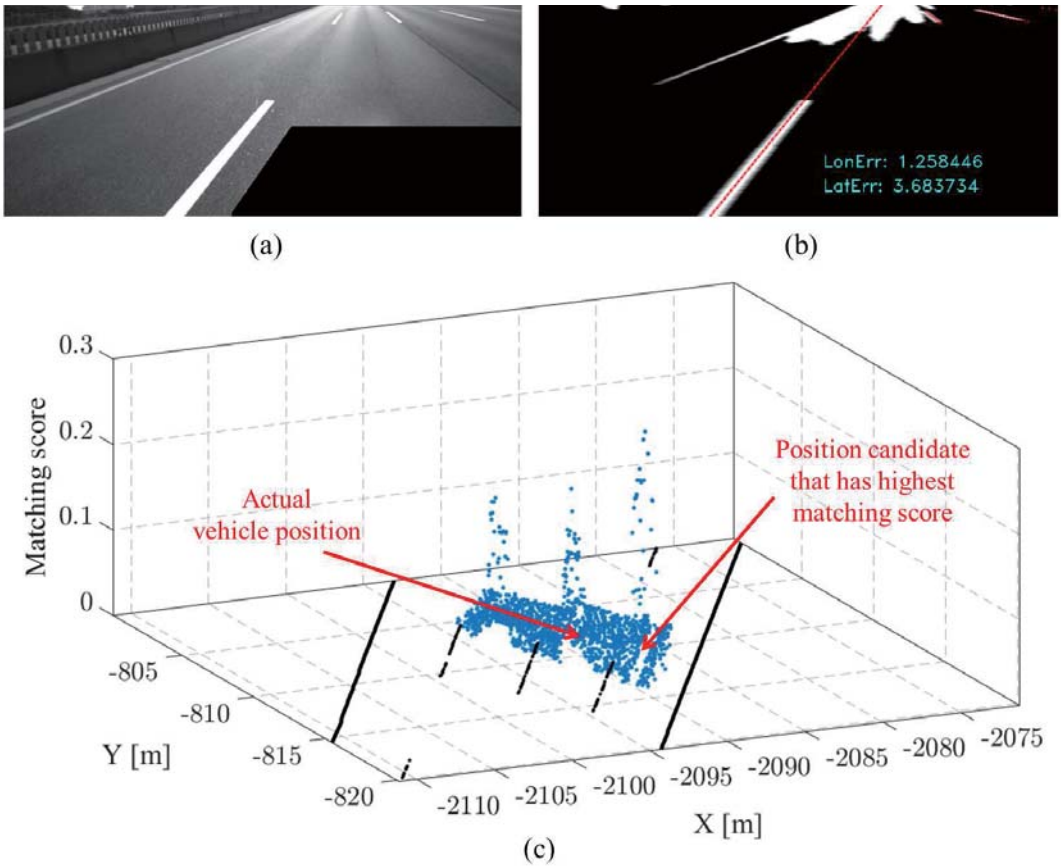


Figure 3.6 Example of the matching ambiguity problem.

to predict the motion of particles at the time-update step. The road map and image are the key information for accurate localization. The lane marking and road marking information contained in the map are matched to the information that the image contains, and the particles are updated using the matching results. In the rest of this chapter, detailed localization algorithm is described.

3.2.4 Problem Statement

The core of the proposed localization algorithm is on estimating accurate pose of the vehicle only using simple road marking features (lane markings and other road mark-

ings appearing intermittently) stored in the map. Lack of these feature information may cause an ambiguity problem. At each measurement update step of the particle filter, a camera image and a map data around the vehicle are given. The problem is to find an optimal vehicle pose that maximize the matching score between the image and the map data. If there is abundant feature information in the map, there exist only one pose solution that has significantly high matching score compared to the other pose candidates. However, in our problem, there exist multiple candidates of solution that have similarly high matching scores due to the ambiguity of the road marking feature information. Fig. 3.6 shows an example of this problem. Fig. 3.6(c) shows the matching score with respect to the vehicle pose candidates distributed around the actual vehicle position, and Fig. 3.6(a) and Fig. 3.6(b) show the map-image matching result for the pose candidate that has highest matching score. As we can see from this example, the ambiguity in the map-image matching can cause large errors that cannot be acceptable for automated driving.

The conventional particle filter cannot handle this ambiguity problem, since it doesn't care of the possibility that the measurement was distorted by the ambiguous problem. In this context, this chapter proposes a particle filter with modified likelihood function to handle the ambiguity problem.

3.3 Particle filter-based Vehicle Localization Algorithm

Fig. 3.7 shows the primary process of the localization algorithm based on the particle filter. At the initialization step, the state and weight values of the particles are initialized based on the GPS positioning information. After finishing the initialization, the recursive filtering process that is composed of the time update and measurement update is performed. At the time update step, the probabilistic distribution of the

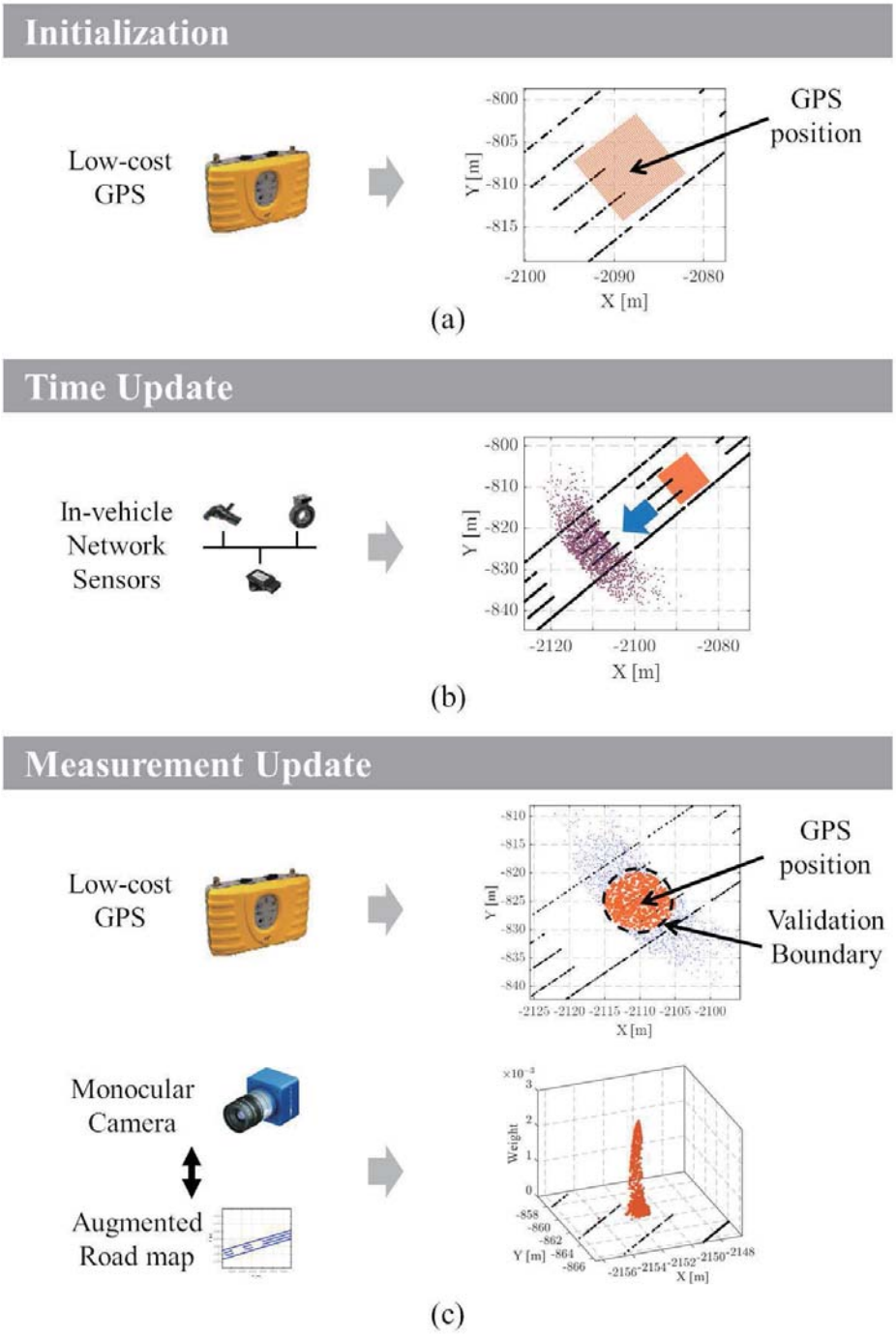


Figure 3.7 Primary particle filter process for vehicle localization.

vehicle pose is predicted by updating the state values of the particles according to the vehicle motion model. At the measurement step, the GPS and map-image measurements are used to update the posterior distribution of the vehicle pose.

3.3.1 Initialization

The initialization is performend based on the GPS positioning information. If an initialization command is issued, the initialization module wait until the healthy GPS positioning information is received. Healthy status of the GPS information is checked according to various GPS signal condition parameters such as number of satellites and horizontal dilution of precision (HDOP). If the status of the GPS information is healthy, initial particles are generated around the GPS position. I assume that the initial vehicle pose $p(x_{0|0})$ is uniformly distributed. According to this assumption, the particles are uniformly generated with rectangular-shape around the GPS position as shown in Fig. 3.7(a), and the weights of the particles are uniformly normalized as follows:

$$w^i = \frac{1}{N}. \tag{3.3}$$

The size of the rectangle depends on the error characteristics of the GPS receiver. I found that 9 m by 9 m size of rectangle have worked well in our experiment.

3.3.2 Time Update

In the time update step, the state of each particle is updated based on the simple vehicle motion model as

$$\begin{aligned} x^i_{k|k-1} &= \begin{bmatrix} \psi^i_{k|k-1} \\ X^i_{k|k-1} \\ Y^i_{k|k-1} \end{bmatrix} \\ &= \begin{bmatrix} \psi^i_{k-1|k-1} + \Delta T \cdot r^i_{k-1} \\ X^i_{k|k-1} + \Delta T \cdot V^i_{k-1} \cos\left(\psi^i_{k-1|k-1} + \frac{1}{2}\Delta T \cdot r^i_{k-1}\right) \\ Y^i_{k|k-1} + \Delta T \cdot V^i_{k-1} \sin\left(\psi^i_{k-1|k-1} + \frac{1}{2}\Delta T \cdot r^i_{k-1}\right) \end{bmatrix}, \end{aligned} \tag{3.4}$$

where r^i and V^i are control inputs corresponding to yaw rate and wheel speed, respectively. The r^i and V^i are random variables rather than deterministic values derived from below error models:

$$\begin{aligned} r^i_k &= r_{\mathrm{GYRO},k} + \varepsilon^i_{\mathrm{GYRO}}, \\ V^i_k &= V_{\mathrm{WHEEL},k} + \varepsilon^i_{\mathrm{WHEEL}}, \end{aligned} \tag{3.5}$$

where $r_{\mathrm{GYRO},k}$ and $V_{\mathrm{WHEEL},k}$ are yaw rate and wheel speed sensor values obtained from the in-vehicle network, and $\varepsilon^i_{\mathrm{GYRO}}$ and $\varepsilon^i_{\mathrm{WHEEL}}$ are corresponding Gaussian errors.

3.3.3 Measurement Update

The weight of each particle w^i_k is updated by evaluating the likelihood of measurements $p(z_k|x^i_{k|k-1}, m)$ conditioned on the time-updated particle state and the map as follows:

$$w^i_k \propto w^i_{k-1} p(z_k|x^i_{k|k-1}, m). \tag{3.6}$$

There are two types of measurements to update the state: GPS position information and a map-image matching measurement. The GPS measurement is responsible for preventing the localization error from diverging. Accurate state correction is performed based on the map-image measurement. This section describes details of the two measurement models and update procedures.

GPS Measurement Update

Generally, the error of the low-cost GPS position data can be modeled as a combination of a Gaussian random noise and a bias error as below:

$$\varepsilon_{GPS} = \varepsilon_{\sigma} + \varepsilon_{b}. \tag{3.7}$$

Even though the magnitude of the bias error varies slowly, it is difficult to model and estimate the magnitude and direction of the error. This nature makes it difficult to use the GPS position information as a measurement of a Bayesian filter. However, fortunately, since the maximum magnitude of the bias error is regulated below few meters under healthy GPS signal environment, we can roughly model the GPS error as a circle-shaped uniform distribution with a radius δ_{GPS}. Then the likelihood function with respect to the GPS measurement $y_{\mathrm{GPS,k}}$ and a particle $x^{i}_{k|k-1}$ is defined as follows:

$$q^{i}_{\mathrm{GPS}} = \begin{cases} 1, & \text{if } \left(\mathrm{X^{i}_{k}} - \mathrm{X_{GPS,k}}\right)^2 + \left(\mathrm{Y^{i}_{k}} - \mathrm{Y_{GPS,k}}\right)^2 \leq \delta^2_{\mathrm{GPS}} \\ 0, & \text{otherwise} \end{cases}. \tag{3.8}$$

Map-Image Measurement Update

The likelihood of the map-image measurement q^{i}_{CAM} is obtained by matching the road map data to the camera image based on each particle state as below:

$$q^{i}_{\mathrm{CAM},k} = p(z_k | x^{i}_{k|k-1}, m). \tag{3.9}$$

The likelihood function for the map-image measurement is the most important part of this chapter. Therefore, I will present more detailed algorithms for the measurement update and likelihood function derivation in Section 3.4.

3.3.4 Integration

Since GPS and map-image measurements are independent, using Eqs. 3.6, the weight is updated as follows:

$$w_k^i = w_{k-1}^i \cdot q_{\mathrm{GPS},k}^i \cdot q_{\mathrm{CAM},k}^i. \tag{3.10}$$

After updating the weights, a normalization process is performed as follows:

$$w_k^i = \frac{w_k^i}{\sum_{j=1}^{N} w_k^i}. \tag{3.11}$$

3.3.5 State Estimation

After updating the weights, we can approximate the posterior distribution with respect to the vehicle position as follows:

$$p(x_{k|k}|z_{1:k}, m) \approx \sum_{i=1}^{N} w_k^i \delta(x - x_{k|k-1}^i), \tag{3.12}$$

where $\delta(x - x_k^i)$ is a delta function centered at x_k^i. The most common way to estimate the state value from the posterior distribution is applying the minimum mean square error (MMSE) estimate, which can be approximated by the weighted mean of the particle density distribution, as follows:

$$\hat{x}_{k|k} = \sum_{i=1}^{N} w_k^i x_{k|k-1}^i. \tag{3.13}$$

However, in many cases, the likelihood distribution of the map-image measurement has a multimodal distribution as shown in Fig. 3.6. In a multimodal distribution, it

is better to use a maximum a-posteriori estimate as follows:

$$\hat{x}_{k|k} = \arg\max_{x^i_{k|k-1}} p\left(x^i_{k|k-1}|z_{1:k}, m\right). \tag{3.14}$$

3.3.6 Resampling

The resampling process is conducted to prevent the particles from being concentrated in specific areas. The degree of the particle concentration can be measured by

$$\hat{N}_{\text{eff}} = \frac{1}{\sum_{i=1}^{N} (w^i_k)^2}. \tag{3.15}$$

$\hat{N}_{\text{eff}}$ approximates the effective number of samples. In this dissertation, if $\hat{N}_{\text{eff}} < 2/3N$, the resampling is conducted based on a systematic resampling method. This method is efficient in terms of computational time.

3.4 Map-Image Measurement Update with Probabilistic Data Association

This section presents the map-image measurement update process. The goal of this step is to get a likelihood distribution of the image measurement conditioned on the particle states and the map. In common with the previous Monte-carlo localization algorithms, proposed algorithm calculates the likelihood value with respect to each particle by calculating the matching score between given camera image and the road map data conditioned on the particle state. However, in order to handle the ambiguity problem of the measurement, the probabilistic data association scheme is adopted, and the likelihood values are updated by so-called association probability.

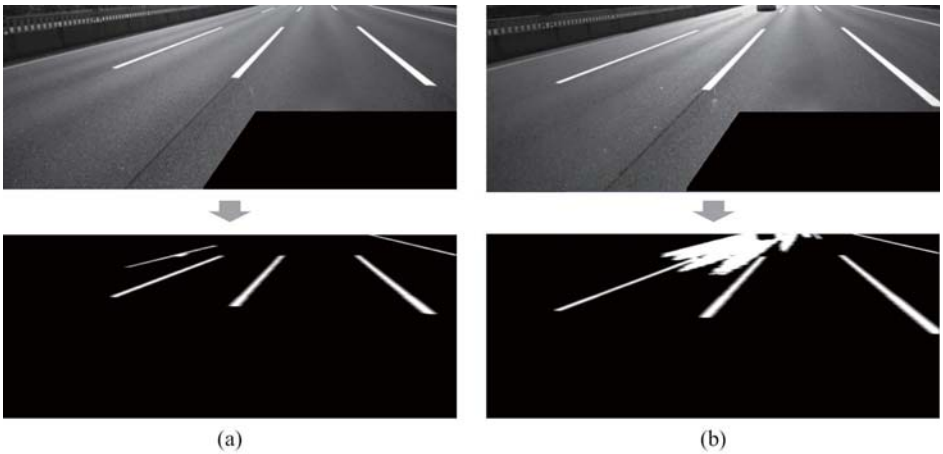


Figure 3.8 Example of the intensity thresholding of images.

3.4.1 Lane Marking Extraction and Measurement Error Model

Road Marking Extraction

In order to match the feature from the camera image to the map data, road marking components are extracted from the camera image. There exist a lot of algorithms for road marking extraction, however, simple intensity thresholding method is used in this dissertation as shown in Fig. 3.8. The reason why simple intensity thresholding method is used is as follows: 1) By simplifying the image processing, we can reduce overall computational time. 2) The identification of the road markings is not needed for the matching algorithm. 3) Some false positives and false negatives are acceptable for the matching algorithm.

Measurement Error Modeling

In this dissertation, only three vehicle pose values (X, Y, and heading) are estimated by assuming that the height, pitch, and roll are constant. This is because the required number of the particles should be exponentially increased by the dimension of the

state vector. In order to reduce the dimension of the state vector, first it is assumed that the road is on a flat plane. This means that the height is constant. However, it is difficult to set the pitch and roll of the vehicle as a constant since they vary frequently in dynamic driving conditions. In order to handle the variation of the pitch and roll angles, the pitch and roll errors are reflected in the measurement (image) by applying the Gaussian filter to the image as shown in Fig. 3.9.

Map to Camera Matching

After finishing the road marking extraction and applying the Gaussian noise, map to camera matching is performed. First, the road map data is transformed to the image plane with respect to each particle state (pose candidate). The transform contains some matrix multiplications and non-linear arithmetic operations. Therefore, in order to reduce the computational burden for real-time operation, the end-points of the road markings are transformed to the image plane, and then the end-points are interpolated in the image plane as shown in Fig. 3.10.

After finishing the coordinate transformation, the measurement likelihood

$$q^i_{\mathrm{CAM},k} = p(z_k|x^i_{k|k-1}, m) \tag{3.16}$$

is calculated by calculating the matching score as follows:

$$p(z_k|x^i_{k|k-1}, m) \propto \exp\left[\frac{1}{N_{p,j}}\sum_{j=1}^{N_{p,j}} -\left(\frac{(I^i_{\mathrm{Cam},j,k} - I^i_{\mathrm{Map},j,k})^2}{2\sigma_I^2}\right)\right], \tag{3.17}$$

where $I_{\mathrm{Map,j}}$ denotes the intensity value of jth pixel corresponding to the map data, $I_{\mathrm{Cam,j}}$ denotes the intensity of image pixel corresponding to the image pixel coordinates of the I_j, and the $N_{p,j}$ means the number of pixels corresponding to the map data. Here, $I_{\mathrm{Map,j}}$ is set to 255 which is the maximum intensity value. Eqs. 3.17 can

(a) Stdv = 0

(b) Stdv = 11

(c) Stdv = 21

Figure 3.9 Example of the Gaussian filtering.

be derived by assuming that the likelihoods of pixels are independent and follow the Gaussian distribution as below:

$$p(z_k|x^i_{k|k-1}, m) = \prod_{j=1}^{N_{p,j}} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-(I^i_{Cam,j,k} - I^i_{Map,j,k})^2}{2\sigma_I^2}\right). \tag{3.18}$$

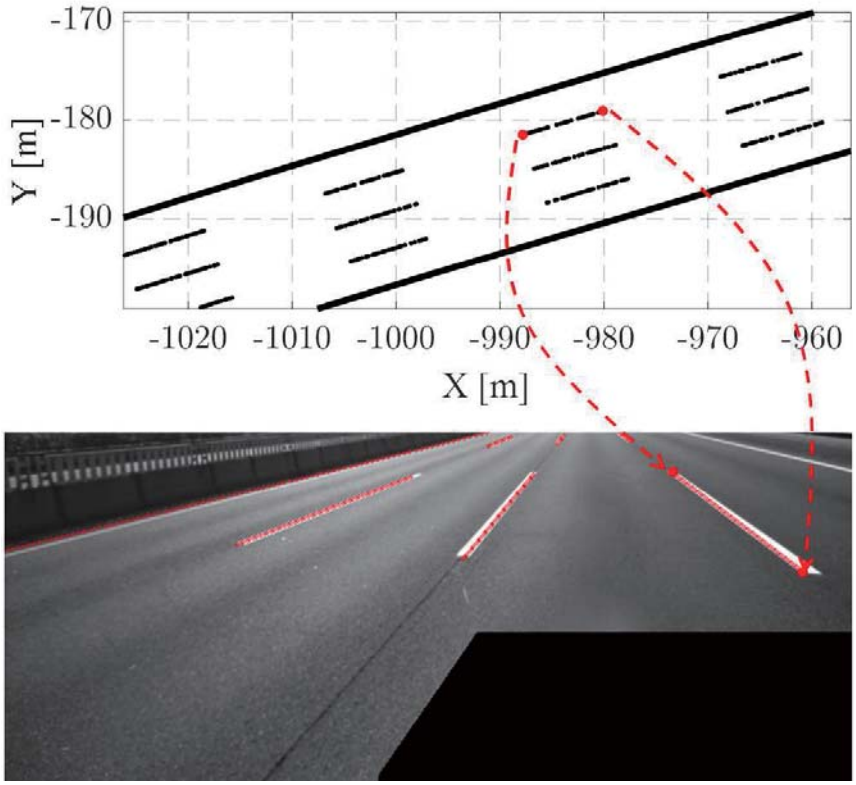


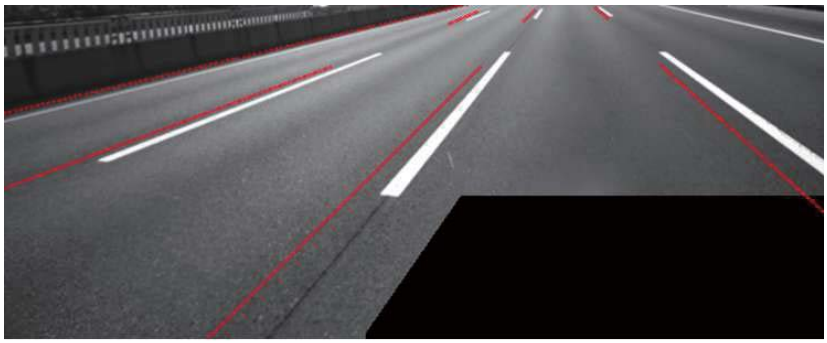
Figure 3.10 Example of the map to image plane transformation.

The likelihood function 3.17 is intuitive in terms of the fact that the likelihood value is equal to 1 if the map data and the image features are perfectly matched, and close to 0 if the map data and the image features are not matched at all.

Fig. 3.11 shows the example of the map-image matching results and corresponding likelihood values for three different particles.

Measurement Association

The measurement model and the likelihood function explained in the previous sections is vulnerable to the measurement ambiguity problem. This section presents the measurement association scheme to handle the ambiguity problem.



(a) Likelihood value = 0.0247

(b) Likelihood value = 0.341

(c) Likelihood value = 0.160

Figure 3.11 Example of the Gaussian filtering.

We can think that the likelihood function obtained in the previous section is based on the hypothesis that all the measurements certainly came from the true state.

Consequently, the likelihood function ignores the hypothesis that the measurement came from false state due to the ambiguity problem. In order to tackle this problem, two mutually exclusive hypotheses are considered as follows:

- θ_0^i : The hypothesis that the measurement is not associated with ith particle.
- θ_1^i : The hypothesis that the measurement is associated with ith particle.

Using these events and the total probability theorem, we can write $p(z_k|x_{k|k-1}^i, m)$ as

$$p(z_k|x_{k|k-1}^i, m) = \sum_{j=0}^{1} p(z_k|x_{k|k-1}^i, m, \theta_{j,k}^i) \cdot p(\theta_{j,k}^i|x_{k|k-1}^i, m). \tag{3.19}$$

Conditioned on the association event we can evaluate $p(z_k|x_{k|k-1}^i, m, \theta_{j,k}^i)$ as

$$p(z_k|x_{k|k-1}^i, m, \theta_{j,k}^i) = \begin{cases} 0 & , j = 0, \\ \exp\left[\frac{1}{N_{p,j}} \sum_{j=1}^{N_{p,j}} -\left(\frac{(I_{\text{Cam},j,k}^i - I_{\text{Map},j,k}^i)^2}{2\sigma_I^2}\right)\right] & , j = 1. \end{cases} \tag{3.20}$$

Here, for simplicity, it is assumed that, for the false detection event, the likelihood is zero.

$\beta_{j,k}^i = p(\theta_{j,k}|x_{k|k-1}^i, m)$ is named as the association probability in this dissertation. Interesting point is that the association probability is independent to the measurement z_k. It is only conditioned on the particle state $x_{k|k-1}^i$ and the map data m. Additionally, I assume that the measurement association event is independent to the map data, and then the association probability can be written as

$$\beta_{j,k}^i = p(\theta_{j,k}^i|x_{k|k-1}^i). \tag{3.21}$$

Since I assumed that the likelihood is zero for the false detection event, the association probability for true detection, $\beta_{1,k}$, is only our interest. $\beta_{1,k}$ can be expressed as

$$p(\theta_{1,k}|x_{k|k-1}^i) = \frac{p(x_{k|k-1}^i|\theta_{1,k}^i)p(\theta_{1,k}^i)}{p(x_{k|k-1}^i)} \tag{3.22}$$

by the Bayes theorem. Here, the prior probability $p(\theta^i_{1,k})$ is constant over the particles, and $p(x^i_{k|k-1})$ are i.i.d, we can conclude that

$$p(\theta_{1,k}|x^i_{k|k-1}) \propto p(x^i_{k|k-1}|\theta_{1,k}) \sim \mathcal{N}(x^i_{k|k-1}; x_{k|k-1}, Q_k). \tag{3.23}$$

Finally, the likelihood of map-image measurement is calculated by

$$p(z_k|x^i_{k|k-1}, m) = \exp\left[\frac{1}{N_{p,j}}\sum_{j=1}^{N_{p,j}}\left(\frac{(I^i_{\text{Cam},j,k} - I^i_{\text{Map},j,k})^2}{2\sigma_I^2}\right)\right] \cdot \mathcal{N}(x^i_{k|k-1}; x_{k|k-1}, Q_k). \tag{3.24}$$

3.5 Experimental Validation

3.5.1 Experimental Environments

Test Road

The proposed localization system was evaluated at the Incheon international airport express (IIAE) in Korea. From the airport tollgate to Geumsan interchange, about 13 km section was selected for evaluation as shown in Fig. 3.3. The IIAE does not contain enough features for localization except the lane marking information. Therefore, it is appropriate to evaluate the proposed localization system. In addition, IIAE includes an about 3 km of solid line road section, which is useful for evaluating the longitudinal localization performance.

Algorithm Implementation

The proposed particle filter-based localization was implemented using 1600 particles. The execution time is about 20∼30 ms based on Intel i7 CPU computing unit. I found that more than 1600 particles do not improve the localization performance much.

Monte-Carlo Experiments

Since the proposed localization algorithm is based on the random sampling, the result may different for each trial. Therefore, it is not reliable to evaluate the performance of localization by using the results of only one of the experiments. To reliably evaluate the algorithm based on random sampling, the evaluation uses the Monte-Carlo method, which repeats the experiment several times with different random samples and the same control inputs. In this dissertation, 10 trials for each experiment were executed, and the results are averaged to get final results.

3.5.2 Localization Accuracy

Comparison to GPS+DR localization

In order to evaluate the performance of the proposed localization, first, a comparison study performed with the GPS+DR-based localization. The GPS+DR-based localization algorithm integrates the low-cost GPS position information and the in-vehicle network sensors using an extended Kalman filter. A RTK-GPS with centimeter-level accuracy was used to obtain a ground-truth vehicle position data, and the localization errors were evaluated based on the ground truth data. Position error, longitudinal position error, and lateral position errors of two algorithms were evaluated, and these errors were obtained by averaging 10 trials of Monte-Carlo experiments. Fig. 3.12 shows the comparison result. We can see that the longitudinal errors are maintained under 1 m in normal conditions, however, in the section between $t = 200$ and $t = 300$, the error get increased remarkably. This section is solid lane marking section in which the longitudinal features do not exist. However, the maximum error is limited below 4 m. This is because the GPS measurement update prevents the error from diverging. We can see the excellence of the proposed algorithm in terms of the lateral local-

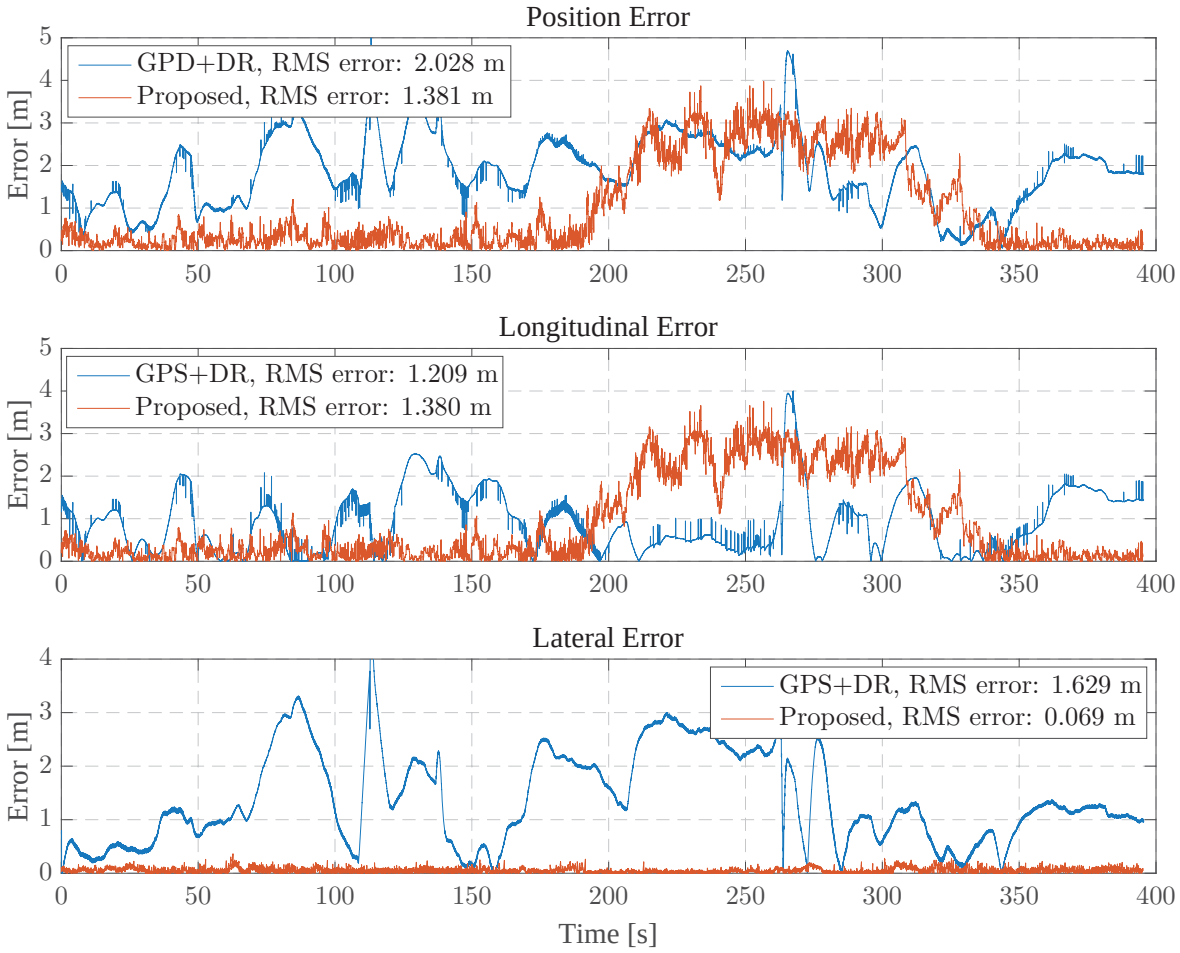


Figure 3.12 Error comparison result of a GPS+DR-based localization algorithm and the proposed localization algorithm.

ization accuracy. The proposed algorithm provides very accurate lateral localization accuracy with a RMS error of about 7 cm. Therefore, from the experiment result, we can see that the proposed algorithm can provide accurate and reliable positioning information which is sufficient for autonomous vehicle driving.

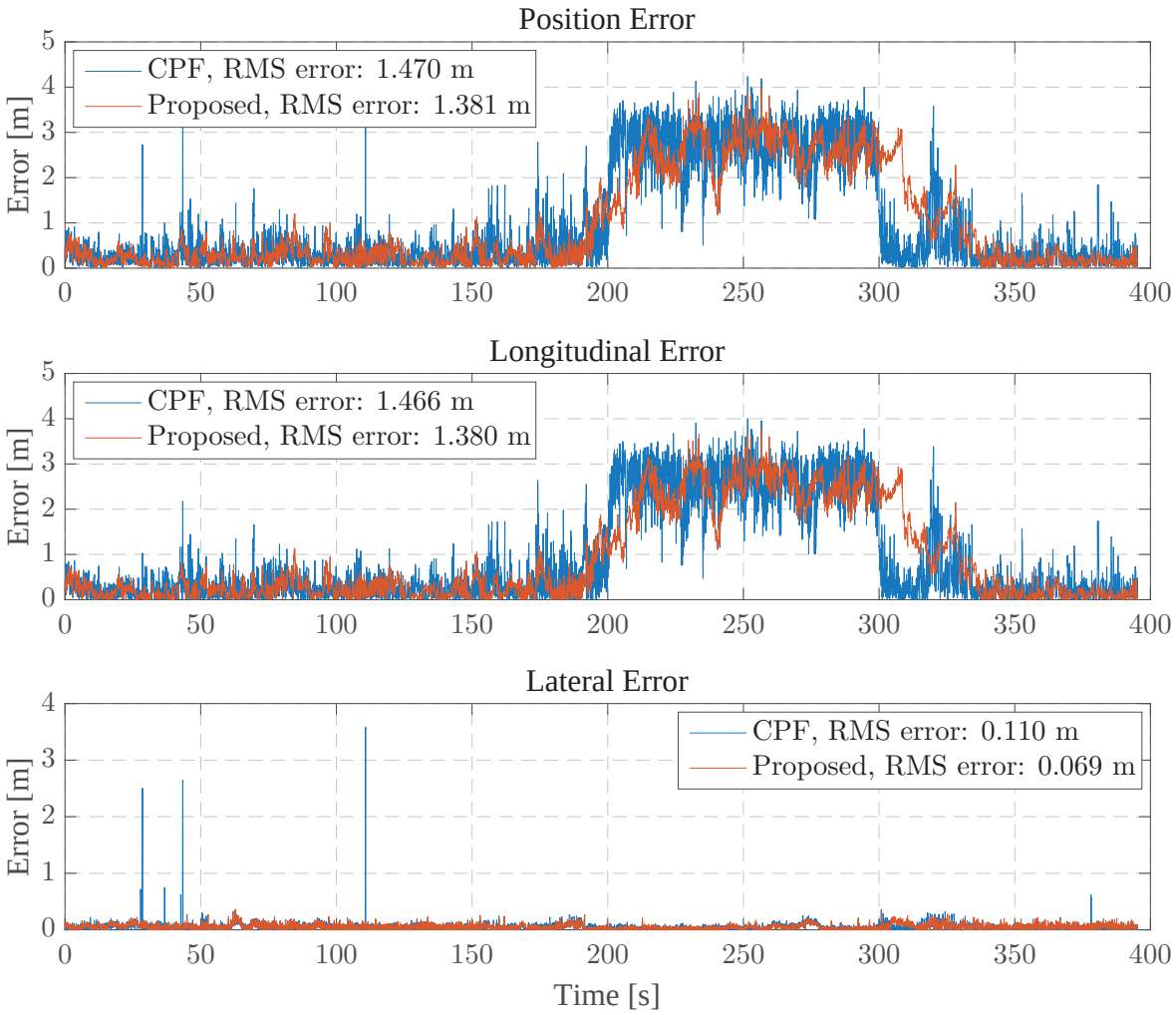


Figure 3.13 Error comparison result of a CPF-based localization algorithm and the proposed localization algorithm.

3.5.3 Effect of the Probabilistic Measurement Association

In order to evaluate the performance of the proposed localization algorithm, a comparison study was performed with conventional particle filter (CPF) with a conventional Gaussian distribution-based likelihood function. Fig. 3.13 shows the error comparison result of the CPF-based localization system and the proposed localization system. One

remarkable point is that, occasionally, the lateral error of the CPF-based localization system increased significantly. These large lateral errors are caused by the failure in map-image matching process due to the measurement ambiguity. As we can see from the result, in case of the proposed algorithm, the lateral errors do not exceed 0.4m. From this result, we can see that the proposed algorithm effectively solves the measurement ambiguity problem and provides accurate and reliable lateral localization result.

3.5.4 Effect of the Measurement Error Model

Table 3.4 The localization performance for various standard deviation values of the Gaussian filter.

STDV [pixel]	0	11	23	35	47
Longitudinal Error [m]	1.578	1.444	1.427	1.435	1.451
Lateral Error [m]	0.742	0.710	0.069	0.708	0.721

The Gaussian noise was applied to the image to handle the variations in the pitch and roll angles. In order to see the effect of the Gaussian noise, experiments were conducted by varying the standard deviation of the Gaussian filter. Table 3.4 shows the result of the experiment. From the results, we can know that an optimal standard deviation value of the Gaussian filter exists. In our experiment, when the standard deviation was set to 23, the localization system shows the best result.

3.6 Summary

This chapter proposed a precise vehicle localization algorithm with poor feature information based on the fusion of a high-precision lane-level road map, a low-cost GPS,

in-vehicle network sensors, and a monocular camera. First, the road map presented in chapter 2 was augmented for including the dashed lane marking information and other road marking information. The augmented road map and other low-cost sensor information were integrated by a particle filter with probabilistic measurement association. The main contribution of this chapter can be summarized as follows:

- The proposed localization algorithm was designed to provide accurate vehicle pose information under the environments where feature information for localization is poor, e.g., highway. The localization system primarily uses the lane marking information, and utilizes sparsely existing road marking information.

- The localization system operates based on low-cost sensors including a low-cost GPS, in-vehicle network sensors, and a monocular camera.

- In order to handle the measurement ambiguity problem due to the poor feature information and use of low-cost sensor configuration, a modified likelihood function was proposed. From the experiment, it is demonstrated that the proposed localization system provides precise vehicle pose information for automated driving.

Chapter 4

An Application of Precise Vehicle Localization: Traffic Flow Enhancement Through the Sharing of Accurate Position Information Among Vehicles

Because of its adverse effects on economy and environment, there have been extensive efforts to resolve the traffic jam problem. However, in spite of the lots of research efforts, the traffic jam is still has considered as a serious transportation problem. In recent years, with the development of communication technologies, a number of studies have attempted to improve the traffic flow through the information exchange among vehicles or provision of useful information about road conditions to vehicles.

Platooning is one of the representative approaches for traffic flow enhancement. In the platooning system, vehicles drive by forming a small group named as the platoon for the purpose of increasing the road capacity that is defined as the maximum traffic flow which is obtainable on a given roadway [69,70]. In a platoon, since vehicles

maintain very short inter-vehicle distance, the road capacity is increased. For a platooning, vehicles are automatically controlled by their own controllers which use the information from the other vehicles within the platoon such as positions, velocities and accelerations. As shown in [71], the road capacity increases by a factor of four in the case where the average platoon size is 20 (number of group vhehicles), it is obvious that the platooning can significantly improve the traffic flow. However, for platooning, since not only all vehicles should be equipped with the automatic controllers work on a same algorithm but also complete safety should be secured, it is not realistic for the present.

More realistic approach is a passive method that assists drivers in behaving more intelligent by providing various useful information through communications. For example, providing a congestion degree of roads helps a driver can choose the optimal route. Another example is providing information about conditions of the roads and position of adjacent vehicles to drivers for the purpose of helping drivers' intelligent behaviors. We can intuitively think that the extension of drivers' view will make them driving more intelligently by anticipating the motions of other vehicles, and thus the overall traffic flow can be improved. However, it is difficult to analyze the effect of driver's behavior on the overall traffic flow. The complexity of the traffic flow makes the analysis difficult.

Instead of an analytical way, through a numerical experiment, I show that, as drivers see more headway of preceding vehicles, the overall traffic flow is improved and the traffic jam is suppressed. I assume that each vehicle c_i can know its exact global position using its own localization system, and transmit it to the following vehicles through a V2V (Vehicle to Vehicle) or V2II (Vehicle-to-Infrastructure) communications. The issues with respect to the communication methods and protocols

are out of the scope of this dissertation. Numerical experiments are performed based on the SOV (stochastic optimal velocity) model [72] that belongs to the category of the cellular automata traffic model [73]. I extend the SOV model considering the extension of drivers' view and compare the traffic flow observed on this extended-SOV (ESOV) model with the traffic flow observed on the original SOV model. The experimental result shows that the sharing of position information among vehicles can significantly improves the traffic flow and suppresses the traffic jam.

Eventually, The contribution of this chapter is summarized as follows:

- I propose an traffic model extended from the SOV model by considering the sharing of accurate position information among vehicles, which allows drivers to know the multiple headways of preceding vehicles.

- Based on the ESOV model, I examine the effect of the position information sharing these information on the traffic flow.

- Consequently, I verify that the sharing of the position information makes the traffic flow more smooth.

This chapter is organized as follows: in section 4.1, the Extended SOV model is proposed after the SOV model is explained with basic simulation results. In section 4.2, the results of the numerical experiments are presented and discussed. Finally, the summary is presented in section 4.3.

4.1 Extended SOV Model

In this section, I present an extended SOV model (ESOV) to examine the effect of accurate vehicle localization and sharing of the location information among the

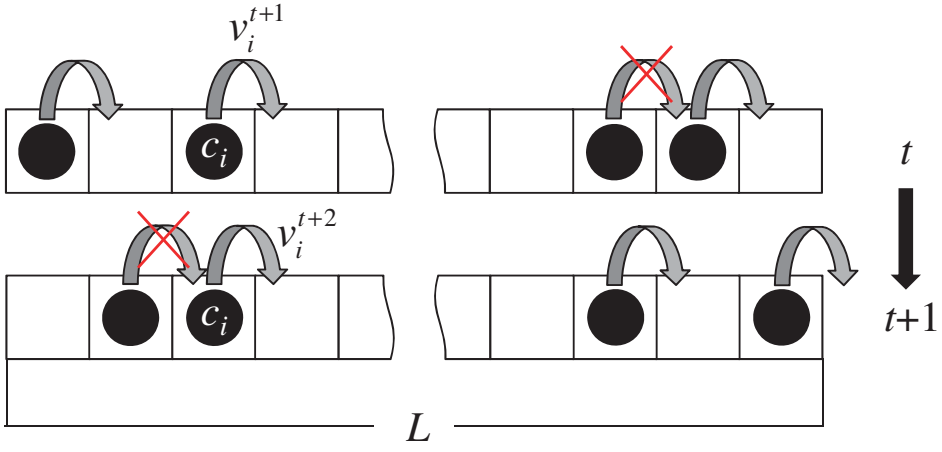


Figure 4.1 The SOV model. A vehicle i hop to the next cell with the probability v_i^{t+1} at time t. Vehicles whose next cell is occupied do not hop.

vehicles. First the original SOV model is explained in section 4.1.1, and then the extended model is explained in section 4.1.2.

4.1.1 SOV Model

This subsection describes the SOV model and presents the reason why I adopted the model. In the SOV model, N vehicles move on a single-lane which is divided into L cells as shown in Fig. 4.1. Each cell contains one vehicle at most, and collision is thus avoided. The positions of vehicles are updated in parallel at each step, i.e., all vehicles attempt to move forward. The velocity v_i^{t+1} of vehicle i at step t is defined by

$$v_i^{t+1} = (1-\alpha)v_i^t + \alpha V_i(\Delta_i), \tag{4.1}$$

where $\alpha(0 \leq \alpha \leq 1)$ is a parameter, Δ_i is the headway, and function V_i is the optimal velocity function, which is defined as

$$V_i(\Delta_i) = \frac{\tanh(\Delta_i - \beta) + \tanh\beta}{1 + \tanh\beta}. \tag{4.2}$$

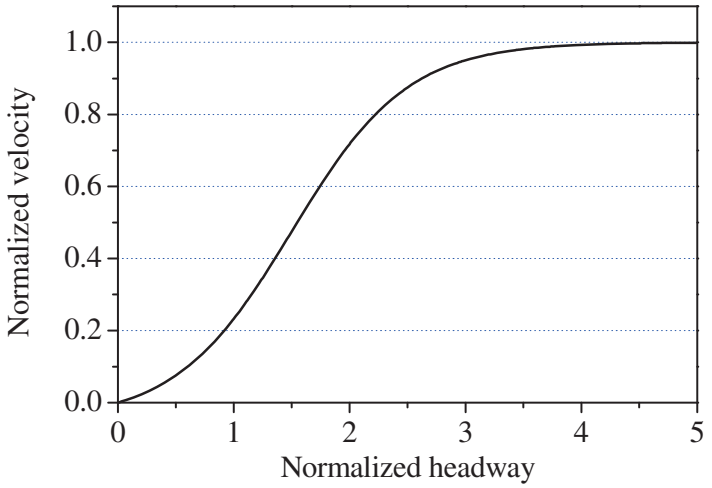


Figure 4.2 The optimal velocity function. The optimal velocity is 0 if the headway is 0 and converges to 1 as the headway increases.

Here, β is a parameter. The function V_i is plotted in Fig. 4.2, where $\beta = 1.5$. V_i reflect the reaction of the vehicle to the headway. Since V is restricted to $[0, 1]$, v_i^t is also restricted to $[0, 1]$, if v_i^0 is less than 1. Note that the normalized velocity v_i^t expresses the hop probability in this model. At each step, each vehicle hop to the next cell with the probability v_i^{t+1} except the case where the next cell is occupied. This is the reason why this model is names as stochastic OV model. Since v_i^{t+1} is the average of v_i^t and V_i, from the Eq. (4.1), variation of the hopping probability of a vehicle depends on its headway.

In spite of its simplicity, the SOV model is plausible for tracing the real traffic flow features. Fig. 4.3 shows the fundamental diagrams of the SOV model with $\alpha = 0.01$ in Eq. (4.1) and $\beta = 1.5$ in Eq. (4.2) for four iteration numbers. In our simulation, the length of the road L was set to 1000 cells and a cyclic boundary condition is applied, i.e., the last cell is linked to the first cell. At the initial step $t = 0$, vehi-

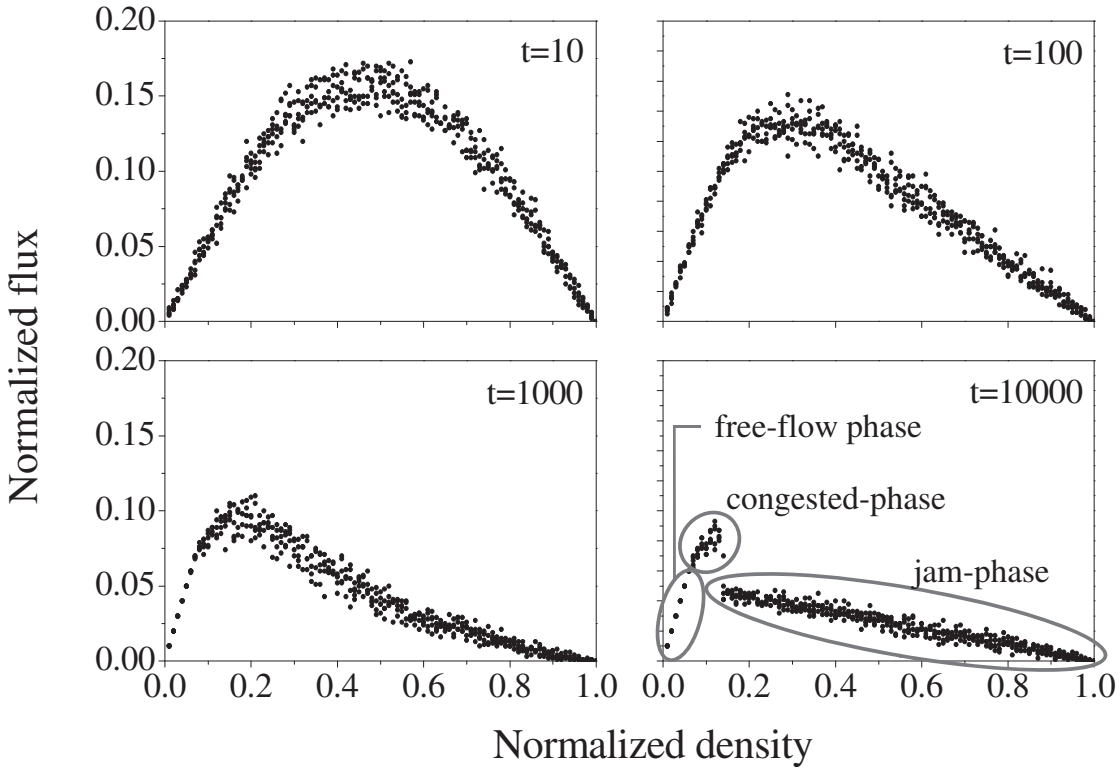


Figure 4.3 The fundamental diagram of the SOV model for $t = 10, 100, 1000$ and 10000. The parameters are $\alpha = 0.01$ and $\beta = 1.5$, and the length of the road is L.

cles are uniformly distributed and velocities are set to $v_i^0 = 0$. The cyclic boundary is appropriate for presenting the homogeneous road which has no bottlenecks. The fundamental diagram shows the variation of the flux according to the density, where the flux is defined by the product of the velocity and density, and thus presents the number of vehicles that pass the unit length per unit time. As time passed, the flux begins to decrease, and finally, the traffic reaches to a stationary state in which the traffic jam exist.

An interesting point is that a discontinuous point appears in the fundamental diagram of stationary state ($t = 10000$) and three different phases are observed around

that point: free-flow phase, congested phase, and jam phase. In the free-flow phase, vehicles move without any interruptions. Therefore, the flux of the free-flow phase is same to the density, where the maximum velocity is 1. In the congested phase, some vehicles are clustered and the average velocity and the flux thus decrease. During the congested phase the small clusters of vehicles are coexist with the free vehicles. A traffic jam is formed in the jam phase. We can see the transition processes of phases through Fig. 4.4. Fig. 4.4-(a) and (b) show the simulation results that present the time evolution of the flux and the distribution of the vehicles at each step respectively, where the density ρ is 1.4. We can know that a traffic jam occurs at the density $\rho = 1.4$ in a stationary state from Fig. 4.3. These two diagrams show the transient process from the initial state to the stationary state. In Fig. 4.4-(a), the phase transition points are clearly appeared. Since I set the initial velocities to v_i^0=1, the traffic start at a free-flow phase. The free-flow phase is lasted until about $t = 5000$ and then replaced by the congested phase. The congested phase is broken at about $t = 10000$, and from this time, the jam phase is lasted. The spatio-temporal diagram which shown in Fig. 4.4 shows the physical features of the congested and jam phase by presenting the position of vehicles at each iteration number. The Y-axis of the diagram denotes the positions of vehicles and the X-axis means the iteration number. In other words, position of each vehicle at each step is denoted as a dot in the diagram, where vehicles move from bottom to up. As shown in the diagram, in the congested phase the vehicles form many small clusters. Relatively darker regions in the diagram present the clustered vehicles. A cluster is produced by a fluctuation of the velocity of a vehicle. If the density of vehicles of the region in which the cluster is formed is high, the cluster becomes a seed of a traffic jam. The diagram shows the forming process of the traffic jam that looks like a black band. Normally, a traffic jam is propagated backward.

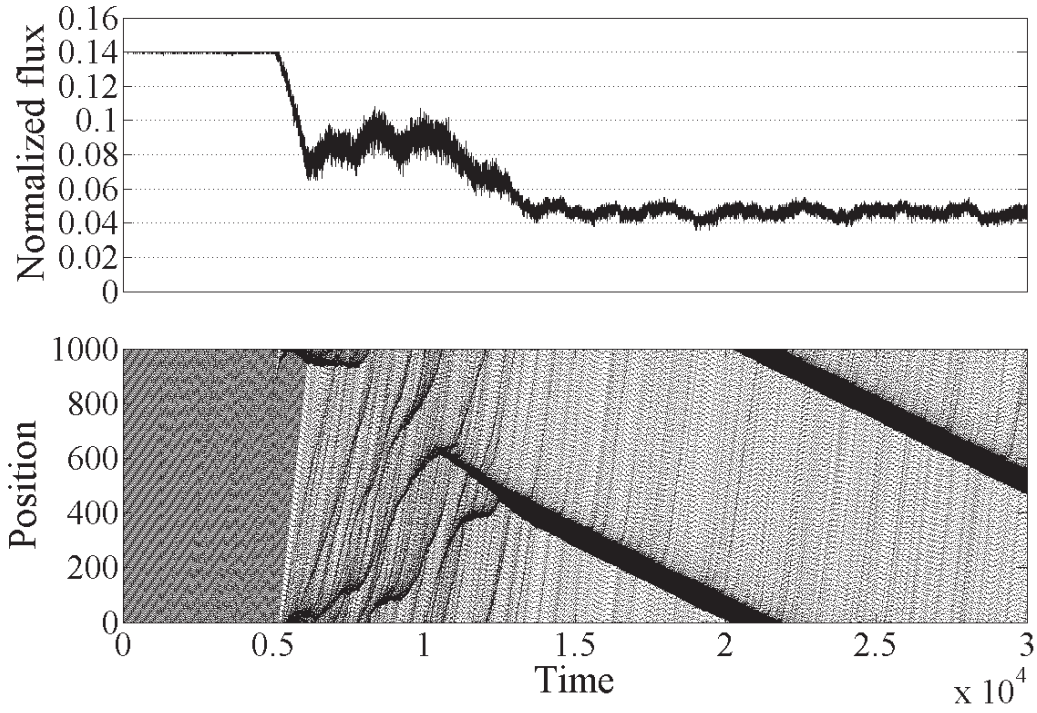


Figure 4.4 (a) The time evolution of flux at the density $\rho = 0.14$, starting from the unifrom state in which all the inter-vehicle distances and velocities of vehicles are identical. The initial velocities are set as $v_i^0 = 1$. (b) The spatio-temporal diagram of the SOV model. A vehicle is denoted as a dot.

The phase transition is important feature that is appeared in the real traffic. This point indicates that the SOV model is realistic and thus led me to adopt this model.

4.1.2 Extended SOV Model

The extended SOV model that I propose considers the intelligent behavior of drivers who receive more information about motions of preceding vehicles. Unlike the original SOV model, the extended model assume that the drivers can see headways of multiple preceding vehicles through the communications, i.e., vehicles receive the headway

or position information from preceding vehicles through the V2V or V2I communications. To reflect this so-called expansion-of-view, I modified the original model without loss of generality as follows:

$$v_i^{t+1} = (1-\alpha)v_i^t + \alpha V_i(w_1\Delta_i + w_2\Delta_{i+1} + ... + w_k\Delta_{i+k-1}), \tag{4.3}$$

where I assume that each vehicle know not only its headway but also the headways of $k-1$ preceding vehicles, and w is weighting factors that satisfies $\sum_{j=1}^{k} w_j$. The extended model is same as the original model when k=1. I give the condition $\sum_{j=1}^{k} w_j$ to make the drivers do the same behavior with the case of the original model when its headway and all the headways of $k-1$ preceding vehicles are identical, i.e., $\Delta_i = \Delta_{i+1}... = \Delta_{i+k}$. Except this condition, I do not suggest a specific rule to determine the weighting factors. I remain it a future work to investigate the effect of weighting factors on the traffic flow. In this dissertation I simply set the weighting factors so that they have same values, i.e., $w_j = 1/k$.

In this model, if the headways of preceding vehicles are relatively long, drivers increase their speed more than the case where the drivers just know its headway (the case of the original model). Similarly, if the headways of preceding vehicles are relatively short, drivers decrease their speed more. In other words, drivers anticipate the near-future motion of preceding vehicles, and they thus accelerate if the preceding vehicles are likely to accelerate even though its current headway is short. The opposite case is the same. Intuitively, we can think that this so-called anticipation effect will make the traffic flow more stable and smooth. As pointed out in [74], a traffic jam can be induced by a tiny fluctuation of the velocity of a vehicle. In the proposed model, since vehicles anticipate the motion of the preceding vehicles, they are less susceptible to the fluctuation of the immediate preceding vehicle. Hence, it is expected that the anticipation effect will suppress the traffic jam. In the next section, I show the result

of the numerical experiment to confirm this hypothesis.

4.2 Results and Discussions

In this section, I present the result of the numerical experiment. First, to ascertain whether the anticipation effect according to the extension of drivers view improves the traffic flow, I examined the effect of the number of front vehicles that each vehicle can perceive on the traffic flow. Fig. 4.5 shows the fundamental diagrams obtained from the numerical experiment based on the extended SOV model. Here, the system size is $L = 1000$ and the cyclic boundary condition was adopted. The graph shows the average flux during the time period of $t = 9000 \sim 10000$ corresponding to the stationary state. Parameters are set as $\alpha = 0.01$ and $\beta = 1.5$ and the initial velocities of vehicles are set as $v_i^0 = 0.5$ identically.

Note that, as k increase, not only the overall average flux increases, but also the formation of the traffic jam is suppressed. From the result, we can know that overall traffic flow is improved about 40% when $k = 3$ and 60% when $k = 5$ than the case where $k = 1$. We also can know that the density value at which the free-flow phase is broken is changed. Whereas the free-flow phase is broken at the density of $\rho = 0.8$ for $k = 1$, it is broken at the density of $\rho = 0.13$ for $k = 3$ and at the density of $\rho = 0.2$ for $k = 5$. Finally, the density at which the traffic jam occurs is also changed. The traffic jam is formed at $\rho = 0.17$ for $k = 1$, $\rho = 0.2$ for $k = 3$, and $\rho = 0.22$ for $k = 5$ respectively.

More detailed features can be seen from the time evolutionary flux diagrams and the spatio-temporal diagrams for $k = 1, 3$ and 5 as shown in Fig. 4.6a, 4.6b and 4.6c respectively. Here, the density is set to $\rho = 0.18$ and the initial velocities are set to $v_i^0 = 0.5$. Even though the results are obtained at the same density value,

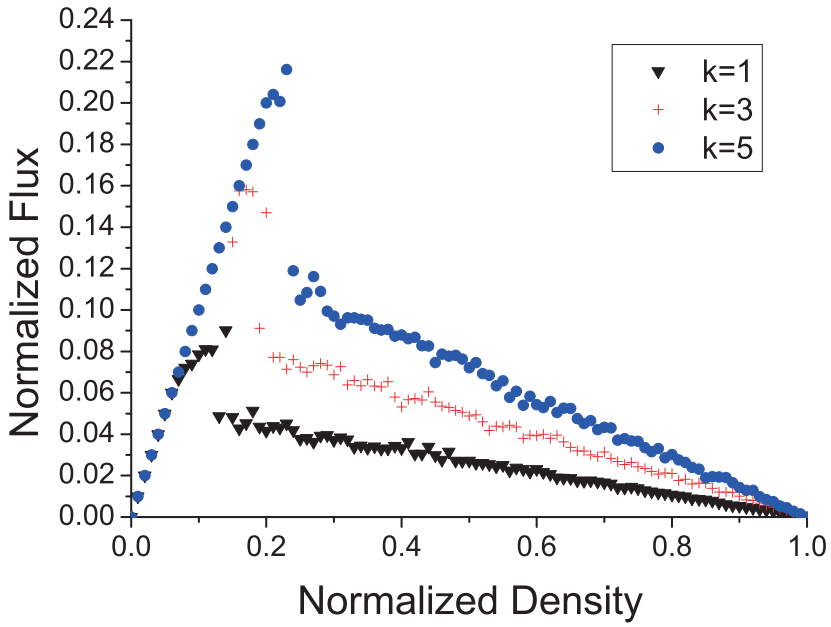
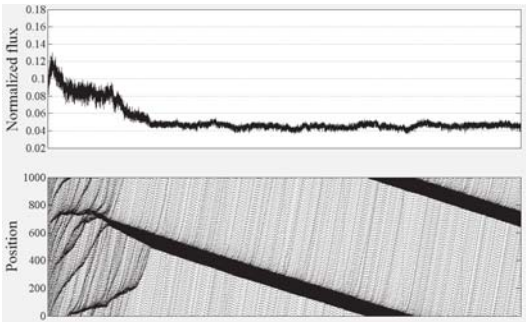
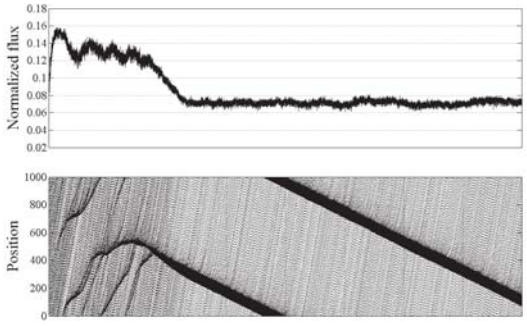


Figure 4.5 The fundamental diagram of the SOV model for $t = 10, 100, 1000$ and 10000. The parameters are $\alpha = 0.01$ and $\beta = 1.5$, and the length of the road L is 1000.

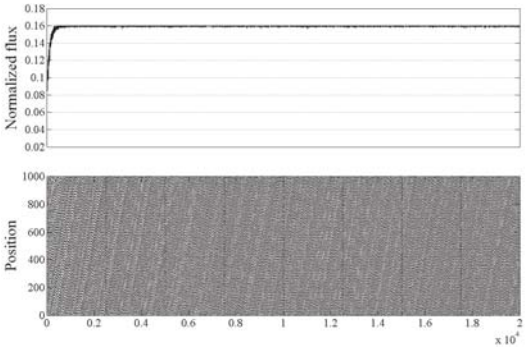
the traffic features shown in diagrams of Fig. 4.6a, Fig. 4.6b, and Fig. 4.6c are fairly different. First, the forming of the cluster is suppressed as k increase. Fewer clusters are observed in the spatio-temporal diagram when $k = 3$ than the case where $k = 1$. One remarkable point is that, for $k = 5$, no clusters and no traffic jam are observed and the traffic is always in the free-flow phase in which the flux is always maximum as shown in Fig. 4.6c. Therefore, we can see that, by the anticipation effect according to the sharing of position information among vehicles, the clusters are dissolved better before growing to the traffic jam. Second, the jam is suppressed. Comparing the spatio-temporal diagram where $k = 3$ with the diagram where $k = 1$, we can see that the width of the jam get narrower and the slope of the jam propagation get steeper as k increases. The narrow width and steep slope of traffic jam means that a vehicle



(a) $k = 1$



(b) $k = 3$



(c) $k = 5$

Figure 4.6 The time evolutionary flux diagrams and the spatio-temporal diagrams. The density is $\rho = 1.8$ and the number of headway that drivers can see are (a) $k = 1$, (b) $k = 3$ and (c) $k = 5$.

that entered the jam area can escape the area more quickly.

4.3 Summary

This chapter demonstrated that the sharing of accurate vehicle position among vehicle can improve the traffic flow and suppress the traffic jam. In order to evaluate the effect of the position information sharing, an ESOV traffic model is proposed by modifying the conventional SOV traffic model. The result of the numerical experiment based

on the ESOV model demonstrates that the traffic flow is increased as a driver can percept the positions of more preceding vehicles.

Chapter 5

Conclusion

In this dissertation I have studied on the mapping and localization problem for automated driving. The accuracy and cost are two primary requirements of the mapping and localization for automated driving. In chapter 2, a road map generation system to generate precise and efficient lane-level road map was proposed. The road map generation system integrates a 3D LIDAR data and high-precision vehicle positioning system to acquire accurate road geometry data. Acquired road geometry data is represented as sets of piecewise polynomial curves in order to increase the storage efficiency and the usability. From extensive experiments based on a real urban and highway road data, I verified that proposed road map generation system generate a road map that is accurate and more efficient than previous road maps in terms of the storage efficiency and usability.

In chapter 3, a precise vehicle localization system was proposed based on the lane-level road map presented in chapter 2, and low-cost sensors. A measurement ambiguity problem due to the use of low-cost sensors and poor feature information was presented,

and a probabilistic measurement association-based particle filter was proposed to resolve the measurement ambiguity problem. Through extensive experiment based on real highway data, the accuracy and reliability of the localization system was evaluated.

In chapter 4, an application of the accurate vehicle localization system was presented. I demonstrated that sharing of accurate position information among vehicles can improve the traffic flow and suppress the traffic jam effectively. The effect of the position information sharing was evaluated based on numerical experiments. For this, the ESOV traffic model was proposed by extending conventional SOV traffic model. The numerical experiments show that the traffic flow is increased as a driver can perceive the position of more preceding vehicles.

I hope this study will contribute to the future of the automated vehicle systems and will be a good foundation for follow-up studies. This study is likely to be expanded in various aspects. The road map generation system proposed in this dissertation operates in a semi-automatic manner. It will be one of good research topics to automatize the road map generation process. The automation of the map generation process is essential to save the cost and time which are needed for replacing current road maps with the map for automated driving. In addition, only the lane information is contained in the map in this dissertation, however, the map will become more valuable if the other additional information is contained in the map such as traffic lights, road signs, and so on. In conjunction with this, the proposed localization system only uses road marking information in the map. However, other landmark information contained in the map can be used to increase the accuracy and reliability of the localization system. I hope many follow-up studies will be carried out and many valuable research results will be produced.

Bibliography

[1] K. Seong-Woo, W. Liu, M. H. Ang, E. Frazzoli, and D. Rus, "The impact of cooperative perception on decision making and planning of autonomous vehicles," *Intelligent Transportation Systems Magazine, IEEE*, vol. 7, no. 3, pp. 39–50, 2015.

[2] C. Urmson, J. Anhalt, D. Bagnell, C. Baker, R. Bittner, M. Clark, J. Dolan, D. Duggins, T. Galatali, C. Geyer *et al.*, "Autonomous driving in urban environments: Boss and the urban challenge," *Journal of Field Robotics*, vol. 25, no. 8, pp. 425–466, 2008.

[3] K. Chu, M. Lee, and M. Sunwoo, "Local path planning for off-road autonomous driving with avoidance of static obstacles," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 13, no. 4, pp. 1599–1616, Dec 2012.

[4] J. Naranjo, C. Gonzalez, R. Garcia, and T. de Pedro, "Lane-change fuzzy control in autonomous vehicles for the overtaking maneuver," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 9, no. 3, pp. 438–450, Sept 2008.

[5] R. Toledo-Moreo, D. Bétaille, F. Peyret, and J. Laneurit, "Fusing GNSS, dead-reckoning, and enhanced maps for road vehicle lane-level navigation," *Selected*

Topics in Signal Processing, IEEE Journal of, vol. 3, no. 5, pp. 798–809, Oct 2009.

[6] R. Toledo-Moreo, D. Bétaille, and F. Peyret, "Lane-level integrity provision for navigation and map matching with GNSS, dead reckoning, and enhanced maps," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 11, no. 1, pp. 100–112, March 2010.

[7] Z. Tao, P. Bonnifait, V. Fremont, and J. Ibanez-Guzman, "Mapping and localization using gps, lane markings and proprioceptive sensors," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on.* IEEE, 2013, pp. 406–412.

[8] N. Suganuma and T. Uozumi, "Precise position estimation of autonomous vehicle based on map-matching," in *Intelligent Vehicles Symposium (IV), 2011 IEEE*, June 2011, pp. 296–301.

[9] M. Schreiber, C. Knoppel, and U. Franke, "Laneloc: Lane marking based localization using highly accurate maps," in *Intelligent Vehicles Symposium (IV), 2013 IEEE*, June 2013, pp. 449–454.

[10] A. Schindler, "Vehicle self-localization with high-precision digital maps," in *Intelligent Vehicles Symposium (IV), 2013 IEEE*, June 2013, pp. 141–146.

[11] E. Consortium *et al.*, "Enhanced digital mapping project. final report." EDMap Project eSafety Forum.(2005). Digital maps Working Group Final Report. European Commission (eSafety Forum), Brussels, 2004.

[12] D. Bétaille and R. Toledo-Moreo, "Creating enhanced maps for lane-level vehicle navigation," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 11, no. 4, pp. 786–798, Dec 2010.

[13] B. Schwarz, "Lidar: Mapping the world in 3D," *Nature Photonics*, vol. 4, no. 7, pp. 429–430, 2010.

[14] K. Jo and M. Sunwoo, "Generation of a precise roadway map for autonomous cars," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 15, no. 3, pp. 925–937, June 2014.

[15] K. Ichida, T. Kiyono, and F. Yoshimoto, "Curve fitting by a one-pass method with a piecewise cubic polynomial," *ACM Transactions on Mathematical Software (TOMS)*, vol. 3, no. 2, pp. 164–174, 1977.

[16] M. Plass and M. Stone, "Curve-fitting with piecewise parametric cubics," in *ACM SIGGRAPH Computer Graphics*, vol. 17, no. 3. ACM, 1983, pp. 229–239.

[17] M. Haklay and P. Weber, "Openstreetmap: User-generated street maps," *Pervasive Computing, IEEE*, vol. 7, no. 4, pp. 12–18, 2008.

[18] J. Ziegler, P. Bender, M. Schreiber, H. Lategahn, T. Strauss, C. Stiller, T. Dang, U. Franke, N. Appenrodt, C. G. Keller *et al.*, "Making bertha drive—an autonomous journey on a historic route," *Intelligent Transportation Systems Magazine, IEEE*, vol. 6, no. 2, pp. 8–20, 2014.

[19] A. Joshi and M. R. James, "Generation of accurate lane-level maps from coarse prior maps and lidar," *Intelligent Transportation Systems Magazine, IEEE*, vol. 7, no. 1, pp. 19–29, 2015.

[20] Y.-W. Seo, C. Urmson, and D. Wettergreen, "Exploiting publicly available cartographic resources for aerial image analysis," in *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*, ser. SIGSPATIAL '12. New York, NY, USA: ACM, 2012, pp. 109–118.

[21] J. Hu, A. Razdan, J. Femiani, M. Cui, and P. Wonka, "Road network extraction and intersection detection from aerial images by tracking road footprints," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 45, no. 12, pp. 4144–4157, Dec 2007.

[22] M. Amo, F. Martinez, and M. Torre, "Road extraction from aerial images using a region competition algorithm," *Image Processing, IEEE Transactions on*, vol. 15, no. 5, pp. 1192–1201, May 2006.

[23] S. Rogers, "Creating and evaluating highly accurate maps with probe vehicles," in *Intelligent Transportation Systems, 2000. Proceedings. 2000 IEEE*, 2000, pp. 125–130.

[24] A. Schindler, G. Maier, and F. Janda, "Generation of high precision digital maps using circular arc splines," in *Intelligent Vehicles Symposium (IV), 2012 IEEE*, June 2012, pp. 246–251.

[25] S. Schroedl, K. Wagstaff, S. Rogers, P. Langley, and C. Wilson, "Mining GPS traces for map refinement," *Data mining and knowledge Discovery*, vol. 9, no. 1, pp. 59–87, 2004.

[26] F. Caron, E. Duflos, D. Pomorski, and P. Vanheeghe, "GPS/IMU data fusion using multisensor Kalman filtering: introduction of contextual aspects," *Information Fusion*, vol. 7, no. 2, pp. 221 – 230, 2006.

[27] H. Qi and J. Moore, "Direct Kalman filtering approach for GPS/INS integration," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 38, no. 2, pp. 687–693, Apr 2002.

[28] X. Chen, B. Kohlmeyer, M. Stroila, N. Alwar, R. Wang, and J. Bach, "Next generation map making: geo-referenced ground-level lidar point clouds for automatic retro-reflective road feature extraction," in *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 2009, pp. 488–491.

[29] A. Boyko and T. Funkhouser, "Extracting roads from dense point clouds in large scale urban environment," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 66, no. 6, pp. S2–S12, 2011.

[30] A. Mancini, E. Frontoni, and P. Zingaretti, "Automatic road object extraction from mobile mapping systems," in *Mechatronics and Embedded Systems and Applications (MESA), 2012 IEEE/ASME International Conference on*. IEEE, 2012, pp. 281–286.

[31] H. Guan, J. Li, Y. Yu, C. Wang, M. Chapman, and B. Yang, "Using mobile laser scanning data for automated extraction of road markings," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 87, pp. 93–107, 2014.

[32] S. Brummer, F. Janda, G. Maier, and A. Schindler, "Evaluation of a mapping strategy based on smooth arc splines for different road types," in *Intelligent Transportation Systems - (ITSC), 2013 16th International IEEE Conference on*, Oct 2013, pp. 160–165.

[33] A. Schindler, G. Maier, and S. Pangerl, "Exploiting arc splines for digital maps," in *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, Oct 2011, pp. 1–6.

[34] J. Goldbeck and B. Huertgen, "Lane detection and tracking by video sensors," in *Intelligent Transportation Systems, 1999. Proceedings. 1999 IEEE/IEEJ/JSAI International Conference on*, 1999, pp. 74–79.

[35] E. Safra, Y. Kanza, Y. Sagiv, and Y. Doytsher, "Efficient integration of road maps," in *Proceedings of the 14th annual ACM international symposium on Advances in geographic information systems*. ACM, 2006, pp. 59–66.

[36] A. Vatavu, R. Danescu, and S. Nedevschi, "Environment perception using dynamic polylines and particle based occupancy grids," in *Intelligent Computer Communication and Processing (ICCP), 2011 IEEE International Conference on*. IEEE, 2011, pp. 239–244.

[37] S. Sivaraman and M. M. Trivedi, "Dynamic probabilistic drivability maps for lane change and merge driver assistance," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 15, no. 5, pp. 2063–2073, 2014.

[38] V. Gikas and J. Stratakos, "A novel geodetic engineering method for accurate and automated road/railway centerline geometry extraction based on the bearing diagram and fractal behavior," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 13, no. 1, pp. 115–126, March 2012.

[39] H. Park and J.-H. Lee, "B-spline curve fitting based on adaptive curve refinement using dominant points," *Computer-Aided Design*, vol. 39, no. 6, pp. 439 – 451, 2007.

[40] P. Sherar, "Variational based analysis and modelling using B-splines," Ph.D. dissertation, Cranfield University, 2004.

[41] D.-S. Kim, J. Ryu, H.-C. Lee, and H. Shin, "The conversion of a dynamic b-spline curve into piecewise polynomials in power form," *Computer-Aided Design*, vol. 34, no. 4, pp. 337–345, 2002.

[42] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, p. 0278364913491297, 2013.

[43] L. Soomok, K. Seong-Woo, and S. Seung-Woo, "Accurate ego-lane recognition utilizing multiple road characteristics in a bayesian network framework," in *Intelligent Vehicles Symposium (IV), 2015 IEEE*, June 2015, pp. 543–548.

[44] K. Seung-Nam, L. Soomok, H. Junhwa, and S. Seung-Woo, "Multi-lane detection based on accurate geometric lane estimation in highway scenarios," in *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*. IEEE, 2014, pp. 221–226.

[45] H. Junhwa, K. Seung-Nam, and S. Seung-Woo, "Multi-lane detection in urban driving environments using conditional random fields," in *Intelligent Vehicles Symposium (IV), 2013 IEEE*. IEEE, 2013, pp. 1297–1302.

[46] Y. Zhou and A. Toga, "Efficient skeletonization of volumetric objects," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 5, no. 3, pp. 196–209, Jul 1999.

[47] H. Wang, J. Kearney, and K. Atkinson, "Arc-length parameterized spline curves for real-time simulation," *Curve and surface design: Saint-Malo 2002.*, pp. 387–396, 2003.

[48] A. Aldoma, Z.-C. Marton, F. Tombari, W. Wohlkinger, C. Potthast, B. Zeisl, R. B. Rusu, S. Gedikli, and M. Vincze, "Point cloud library," *IEEE Robotics & Automation Magazine*, vol. 1070, no. 9932/12, 2012.

[49] R. Toledo-Moreo, M. Zamora-Izquierdo, B. Úbeda-Miñarro, A. F. Gómez-Skarmeta *et al.*, "High-integrity imm-ekf-based road vehicle navigation with low-cost gps/sbas/ins," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 8, no. 3, pp. 491–511, 2007.

[50] S. Rezaei and R. Sengupta, "Kalman filter-based integration of dgps and vehicle sensors for localization," *Control Systems Technology, IEEE Transactions on*, vol. 15, no. 6, pp. 1080–1088, 2007.

[51] J. Huang and H.-S. Tan, "A low-order dgps-based vehicle positioning system under urban environment," *Mechatronics, IEEE/ASME Transactions on*, vol. 11, no. 5, pp. 567–575, 2006.

[52] Y. Cui and S. S. Ge, "Autonomous vehicle positioning with gps in urban canyon environments," *Robotics and Automation, IEEE Transactions on*, vol. 19, no. 1, pp. 15–25, 2003.

[53] Z. Tao, P. Bonnifait, V. Fremont, and J. Ibanez-Guzman, "Lane marking aided vehicle localization," in *Intelligent Transportation Systems-(ITSC), 2013 16th International IEEE Conference on*. IEEE, 2013, pp. 1509–1515.

[54] T. Wu and A. Ranganathan, "Vehicle localization using road markings," in *Intelligent Vehicles Symposium (IV), 2013 IEEE*. IEEE, 2013, pp. 1185–1190.

[55] H. Lategahn and C. Stiller, "Vision-only localization," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 15, no. 3, pp. 1246–1257, 2014.

[56] M. Schreiber, C. Knoppel, and U. Franke, "Laneloc: Lane marking based localization using highly accurate maps," in *Intelligent Vehicles Symposium (IV), 2013 IEEE*. IEEE, 2013, pp. 449–454.

[57] J. Ziegler, H. Lategahn, M. Schreiber, C. G. Keller, C. Knoppel, J. Hipp, M. Haueis, and C. Stiller, "Video based localization for bertha," in *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*. IEEE, 2014, pp. 1231–1238.

[58] K. Jo, Y. Jo, J. K. Suhr, H. G. Jung, and M. Sunwoo, "Precise localization of an autonomous car based on probabilistic noise models of road surface marker features using multiple cameras," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 16, no. 6, pp. 3377–3392, 2015.

[59] K. Jo, K. Chu, and M. Sunwoo, "Interacting multiple model filter-based sensor fusion of gps with in-vehicle sensors for real-time vehicle positioning," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 13, no. 1, pp. 329–343, 2012.

[60] A. Geiger, J. Ziegler, and C. Stiller, "Stereoscan: Dense 3d reconstruction in real-time," in *Intelligent Vehicles Symposium (IV)*, 2011.

[61] S. Thrun and J. J. Leonard, "Simultaneous localization and mapping," in *Springer handbook of robotics*. Springer, 2008, pp. 871–889.

[62] M. Montemerlo and S. Thrun, "Simultaneous localization and mapping with unknown data association using fastslam," in *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, vol. 2. IEEE, 2003, pp. 1985–1991.

[63] M. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba, "A solution to the simultaneous localization and map building (slam) problem,"

Robotics and Automation, IEEE Transactions on, vol. 17, no. 3, pp. 229–241, 2001.

[64] D. Gruyer, R. Belaroussi, and M. Revilloud, "Map-aided localization with lateral perception," in *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*. IEEE, 2014, pp. 674–680.

[65] N. Suganuma and T. Uozumi, "Precise position estimation of autonomous vehicle based on map-matching," in *Intelligent Vehicles Symposium (IV), 2011 IEEE*. IEEE, 2011, pp. 296–301.

[66] J. Levinson and S. Thrun, "Robust vehicle localization in urban environments using probabilistic maps," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. IEEE, 2010, pp. 4372–4378.

[67] N. Mattern and G. Wanielik, "Camera-based vehicle localization at intersections using detailed digital maps," in *Position Location and Navigation Symposium (PLANS), 2010 IEEE/ION*. IEEE, 2010, pp. 1100–1107.

[68] N. Mattern, R. Schubert, and G. Wanielik, "High-accurate vehicle localization using digital maps and coherency images," in *Intelligent Vehicles Symposium (IV), 2010 IEEE*. IEEE, 2010, pp. 462–469.

[69] D. Swaroop, "String stability of interconnected systems: An application to platooning in automated highway systems," *California Partners for Advanced Transit and Highways (PATH)*, 1997.

[70] D. Swaroop and J. Hedrick, "Constant spacing strategies for platooning in automated highway systems," *Journal of dynamic systems, measurement, and control*, vol. 121, no. 3, pp. 462–470, 1999.

[71] U. Karaaslan, P. Varaiya, and J. Walrand, "Two proposals to improve freeway traffic flow," in *American Control Conference, 1991.* IEEE, 1991, pp. 2539–2544.

[72] M. Kanai, K. Nishinari, and T. Tokihiro, "Stochastic optimal velocity model and its long-lived metastability," *Physical Review E*, vol. 72, no. 3, p. 035102, 2005.

[73] K. Nagel and M. Schreckenberg, "A cellular automaton model for freeway traffic," *Journal de physique I*, vol. 2, no. 12, pp. 2221–2229, 1992.

[74] Y. Sugiyama, M. Fukui, M. Kikuchi, K. Hasebe, A. Nakayama, K. Nishinari, S.-i. Tadaki, and S. Yukawa, "Traffic jams without bottlenecks—experimental evidence for the physical mechanism of the formation of a jam," *New Journal of Physics*, vol. 10, no. 3, p. 033001, 2008.

국문 초록

본 논문은 무인자율주행을 위한 정밀하고 효율적인 지도 생성 및 측위 알고리즘을 제안하는 것을 목적으로 한다. 지도 생성 및 측위는 무인자율주행을 위한 핵심 요소로서, 이를 위한 연구에서 주요하게 고려해야할 점은 정확도 및 정밀도를 최대화 하면서 비용은 최소화 하는 것이다. 이러한 목표를 위해, 본 논문은 정밀하면서 비용 효율적인 차선 수준의 도로 지도를 생성하기 위한 시스템을 제안하고, 생성한 차로 지도와 저가형 센서를 활용한 고정밀 측위 시스템을 제안한다.

2 장에서는 도로 지도 생성 시스템을 제시한다. 제안하는 도로 지도 생성 시스템은 먼저 3 차원 라이다 데이터와 고정밀 측위 장비를 결합하여 정확한 도로 지오메트리 데이터를 취득한다. 다음으로, 도로 지도의 저장공간 효율성 및 활용성을 높이기 위해 취득한 도로 지오메트리 데이터를 구간적 다항식 형태의 곡선 식으로 모델링한다. 실제 도심 및 고속도로 데이터를 이용한 실험을 통해 제안한 도로 생성 시스템이 기존 방식에 비해 저장공간 효율성 및 활용성 측면에서 뛰어난 효율성을 지님을 보인다.

3 장에서는 2 장에서 생성한 도로 지도 및 다양한 저가형 센서를 활용한 자동차 측위 시스템을 제시한다. 제안하는 측위 시스템은 피쳐 정보가 부족하여 기존 측위 기술을 적용하기 어려운 환경에서의 정확한 측위를 목표로 한다. 저가형 센서의 사용 및 부족한 피쳐 정보로 인해 발생하는 이른바 계측값 모호성 문제를 제시하고, 이 문제를 해결하기 위한 확률적 계측값 연계 기반의 입자 필터를 제안한다. 실제 도로 데이터를 이용한 실험을 통해 제안한 측위 시스템의 정확도 및 신뢰도를 평가한다.

4 장에서는 정확한 자동차 측위의 한 응용으로서, 정확한 측위 데이터가 교통 흐름 개선에 기여할 수 있음을 보인다. 차량 간 측위 데이터 교환의 효과를 수치적 실험을 통해 입증하기 위해 기존에 널리 사용되는 SOV 교통 모델에 차량 간 측위 데이터 교환 효과를

추가한 확장 SOV 교통 모델을 제안한다. 제안한 모델을 이용하여 차량 간에 정확한 측위 데이터를 교환함으로써 교통 흐름을 크게 개선할 수 있음을 보인다.

주요어: 지능형 자동차, 도로 지도 생성, 자동차 측위

학번: 2009-20753

Ph.D. DISSERTATION

Road Map Generation and Vehicle Localization for Automated Driving

무인자율주행을 위한 도로 지도 생성 및 측위

권 기 풍

FEBRUARY 2016

DEPARTMENT OF ELECTRICAL ENGINEERING AND
COMPUTER SCIENCE
GRADUATE SCHOOL
SEOUL NATIONAL UNIVERSITY

Road Map Generation and Vehicle Localization for Automated Driving

무인자율주행을 위한 도로 지도 생성 및 측위

지도교수 서 승 우

이 논문을 공학박사 학위논문으로 제출함

FEBRUARY 2016

서울대학교 대학원

전기 컴퓨터 공학부

권 기 풍

권 기 풍의 공학박사 학위논문을 인준함

FEBRUARY 2016

위 원 장	이 경 수	
부위원장	서 승 우	
위 원	김 현 진	
위 원	김 성 우	
위 원	정 한 유	

Abstract

This dissertation aims to present precise and cost-efficient mapping and localization algorithms for autonomous vehicles. Mapping and localization are ones of the key components in autonomous vehicles. The major concern for mapping and localization research is maximizing the accuracy and precision of the systems while minimizing the cost. For this goal, this dissertation proposes a road map generation system to create a precise and efficient lane-level road map, and a localization system based on the proposed road map and affordable sensors.

In chapter 2, the road map generation system is presented. The road map generation system integrates a 3D LIDAR data and high-precision vehicle positioning system to acquire accurate road geometry data. Acquired road geometry data is represented as sets of piecewise polynomial curves in order to increase the storage efficiency and the usability. From extensive experiments using a real urban and highway road data, it is verified that the proposed road map generation system generates a road map that is accurate and more efficient than previous road maps in terms of the storage efficiency and usability.

In chapter 3, the localization system is presented. The localization system targets an environment that the localization is difficult due to the lack of feature information for localization. The proposed system integrates the lane-level road map presented in chapter 2, and various low-cost sensors for accurate and cost-effective vehicle localization. A measurement ambiguity problem due to the use of low-cost sensors and poor feature information was presented, and a probabilistic measurement association-

based particle filter is proposed to resolve the measurement ambiguity problem. Experimental results using a real highway road data is presented to verify the accuracy and reliability of the localization system.

In chapter 4, an application of the accurate vehicle localization system is presented. It is demonstrated that sharing of accurate position information among vehicles can improve the traffic flow and suppress the traffic jam effectively. The effect of the position information sharing is evaluated based on numerical experiments. For this, a traffic model is proposed by extending conventional SOV traffic model. The numerical experiments show that the traffic flow is increased based on accurate vehicle localization and information sharing among vehicles.

Keywords: Road map, roadway map, piecewise polynomial, vehicle localization

Student Number: 2009-20753

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and Motivations

Significant efforts have been made in academia and in industry to develop more intelligent transportation systems (ITSs) that can provide a higher level of safety and convenience. Over the past ten years, starting with the development of various ADASs (advanced driver assistance systems), development of autonomous vehicles has been actively conducted, which is the eventual goal of the ITS research. In order for the automated driving to be realized, a lot of component functions should be supported including environmental perception, vehicle localization and motion planning, etc. Among these functions, vehicle localization is one of the most basic and important component for the autonomous vehicles, which is based on the many other autonomous vehicle functions. For example, vehicle path and motion planning start from accurate vehicle localization and various environmental perception algorithms benefit from the vehicle localization.

The vehicle localization can be defined as a function that provides accurate and precise location coordinates of the vehicle on a map. This definition contains a meaning that it should be preceded to create precise road maps for accurate vehicle localization, and then the localization is performed based on the information stored in the map. In this context, studies on the vehicle localization have been conducted in two aspects: 1) creating an accurate and useful road map (mapping), and 2) developing an accurate and efficient localization algorithm.

In fact, we are frequently experiencing the mapping and localization technologies in our daily lives through the vehicle navigation systems. The vehicle navigation systems have a road map and they provide the information about the vehicle location for drivers. However, conventional vehicle navigation systems just provide meter-level location information, thus they are inappropriate to use for autonomous vehicles. The inaccuracy of the vehicle navigation systems comes from the inaccuracies of both the mapping and localization methods. Generally, a vehicle navigation system uses a road map that only contains simplified road geometry information. It represents the road as a network that consists of nodes (crossways) and links (road sections) rather than expressing lane-level road geometry or more complex road feature information. Furthermore, the vehicle navigation systems use a low-cost GPS receiver for localization. Commonly, a low-cost GPS has a meter-level error and it can be increase to tens of meters according to the surrounding environment.

Actually, the vehicle localization problem gets easier with high-tech equipment such as RTK-GPS, high-end INS, high-end computing and storage systems, etc. Therefore, studies on mapping and vehicle localization have been conducted to reduce required cost while increasing the accuracy and precision. In this context, this dissertation aims to propose cost-effective and precise mapping and localization systems

for autonomous vehicles.

In chapter 2, a mapping system to generate a precise and cost-effective lane-level road map is proposed. It is shown that the map generated by the proposed system has enough accuracy to use for automated driving and highly efficient in terms of storage space and usability. In chapter 3, a localization system is proposed based on the lane-level road map and low-cost sensors including in-vehicle network sensors, low-cost GPS and monocular camera. It is shown that the proposed system provides accurate vehicle positioning information with low-cost sensors under an environment in which feature information for localization is poor. In chapter 4, an application of the accurate localization is presented. It is demonstrated that accurate vehicle localization and inter-vehicle communication can increase the traffic flow significantly.

1.2 Contributions and Outline of the Dissertation

1.2.1 Generation of a Precise and Efficient Lane-Level Road Map

Conventional road maps that are used for vehicle navigation systems or geographical information systems are insufficient to satisfy new requirements of intelligent vehicle systems such as automated driving. However, a highly precise map that satisfies the requirements for intelligent vehicle systems requires large amount of storage and thus has a low level of usability. In chapter 2, I propose a precise lane-level road map system that is suitable for automated driving and also conforms to the required storage efficiency and usability criteria. The proposed map building process consists of three steps: 1) data acquisition, 2) data processing, and 3) road modeling. The road data acquisition and processing system captures accurate 3D road geometry data by acquiring data with a mobile 3D laser scanner. The road geometry data is then refined to extract meta information, and in the road modeling system, the refined data

is represented as the set of polynomial-based spline curves to ensure storage efficiency and usability of the map. The proposed mapping system has been extensively tested and evaluated on a real urban road, and highway. The results of the experiment indicated that the proposed mapping system outperforms conventional ones in terms of storage efficiency and usability.

1.2.2 Accurate and Cost-Effective Vehicle Localization in Featureless Environments

There have been fewer studies on vision-based/map-aided vehicle localization for the environment where feature information for localization is poor, e.g., a highway environment. In chapter 3, I propose a precise and cost-effective vehicle localization system based on four cost-effective data sources including a low-cost GPS, in-vehicle network sensors, a monocular camera, and a high-precision lane-level road map for featureless environments. The combination of the low-cost sensor configuration and the poor feature information causes an ambiguity problem in the map-image matching process for localization. To handle this problem, a probabilistic measurement association-based particle filter is proposed. The proposed localization system has been evaluated on a real highway road. The results of the experiments demonstrate that the proposed localization system effectively solves the measurement ambiguity problem and provides accurate vehicle pose information for automated driving.

1.2.3 An Application of Precise Vehicle Localization: Traffic Flow Enhancement Through the Sharing of Accurate Position Information Among Vehicles

Traffic jam is a serious social problem in the view point of economy and environment. In chapter 4, I show that precise vehicle localization can be utilized to enhance the

traffic flow by sharing the position information among vehicles through a numerical experiment. The stochastic optimal velocity (SOV) traffic model that is well known to model the actual traffic flow correctly is used for the numerical experiment. I proposed an extended SOV (ESOV) model reflecting the effect of the position information sharing while retaining the generality of the conventional SOV model, and demonstrated that the sharing of accurate position information among vehicles can improve the traffic flow significantly using the ESOV model.

Chapter 2

Generation of a Precise and Efficient Lane-Level Road Map

It is necessary to have a precise lane-level road map to provide various autonomous driving vehicle applications, such as vehicle path or motion planning [1–4] and ego-vehicle localization [5–10]. In addition, various advanced driver assistance systems (ADASs) including lane keeping, lane change assist, and fuel management systems, benefit from having precise lane-level road maps [11].

From a technical point of view, a map for intelligent and autonomous vehicle systems has to fulfill the following three requirements: accuracy, storage efficiency, and usability. First, a road map should contain the geometry of all lanes with an accuracy at the centimeter-level as well as the 3D structure of roads in order to represent various types of roads, including sloped roads, overpasses, etc. Second, the geometry of the road should be expressed in a compact form such that the map can be downloaded and updated via wireless networks. Finally, the road geometry should be expressed in an application-friendly format. Specifically, low computational efforts

should be necessary to calculate the road geometry information, including coordinates, tangent angle or curvature from the road map to provide real-time operation.

Conventional road maps only provide macro-scale information for vehicle navigation systems or geographical information systems (GISs), so many studies have been carried out to produce a precise road map intended for use with intelligent vehicle systems. Bétaille *et al.* proposed an Emap structure for meso-scale lane-level driving assistance that utilizes GPS and Dead Reckoning (DR) [12]. However, the quality of the map is easily affected by the data acquisition process and is inadequate for autonomous driving purpose. A map for autonomous driving requires micro-scale precision, but such a precise map requires a large amount of storage space and has a low level of usability [13]. Jo *et al.* proposed a mapping system that can generate a storage-efficient map suitable for autonomous driving purposes without considering lanes, complex junctions and 3D traffic structures such as an overpass and a slope [14].

Few previous studies have considered the latter two requirements while providing centimeter-level functionality. In this chapter, I propose a road map generation system that considers the three aforementioned road map requirements altogether. The road map generation system is composed of three subsystems, including data acquisition, processing, and road modeling systems. The data acquisition and processing system acquires accurate road geometry data that fulfills the accuracy requirement, and I introduce a data acquisition system to obtain highly-accurate 3D road geometry by utilizing 3D mobile laser scanning (3D MLS) data. Then, the road geometry data that is obtained is refined to extract the corresponding meta information.

The main contribution of this chapter lies in the road modeling system. In the road modeling system, the road geometry data that is obtained is approximated and represented as sets of piecewise parametric polynomials in order to conform to the

latter two road map requirements. The piecewise polynomial is more effective in terms of usability than those previously used in other road models, including clothoid or B-spline, because various types of road geometry information such as the tangent angle and the curvature of the road can be calculated by conducting simple arithmetic operations. In addition, the piecewise polynomial form is suitable for local modification of the map, since the modification of the coefficients of a piecewise polynomial does not affect the shape of the other piecewise polynomials.

The main problem in representing the road as the sets of piecewise polynomials is the need to minimize the required data size, since the geometry data about the length of the tens-of-thousands kilometers of roads should be stored in a storage device of a vehicle. However, previous piecewise polynomial-based curve approximation algorithms are inappropriate for processing the significantly large amount of road geometry data, since these global optimization algorithms have exponential or $O(n^3)$ computational complexity [15, 16]. In this chapter, I propose an efficient road modeling algorithm that has $O(n)$ complexity. The proposed algorithm sub-optimally determines the number of piecewise-polynomials of the overall curve and its coefficients by converting the global approximation problem to a combination of small easy-to-solve approximation problems.

The proposed road map generation system has been extensively tested and evaluated on an urban road and a highway, and the results of the experiment show that the proposed map outperforms conventional maps in terms of the road map requirements. Our scheme can thus contribute to building a precise and efficient map for navigation systems that can be embedded in autonomous driving vehicles and updated via wireless networks.

The main contributions of this chapter can be summarized as follows:

- I propose a road map generation system to generate a precise and efficient lane-level road map for intelligent and autonomous vehicle systems.
- I propose an efficient curve approximation algorithm that represents the road as the minimum number of piecewise polynomials.
- The feasibility and practicality of the proposed map were evaluated by conducting extensive experiments on an urban road and a highway.

The remainder of this chapter is organized as follows. Section 2.1 presents related works. Section 2.2 presents the overall system architecture of the road map generation system. Section 2.3 introduces the road geometry data acquisition and processing system, and Section 2.4 presents the proposed road representation system. Section 2.5 provides the experimental results and Section 2.6 summarizes this chapter.

2.1 Related Works

The accuracy, storage efficiency and usability of a map all depend on the combination of data acquisition and road modeling methods that are used. The objective of data acquisition is to obtain accurate data to represent the actual geometry of the road, and the primary purpose of road modeling is to effectively represent the road geometry in terms of the storage efficiency and usability while maintaining a certain level of accuracy. Table 2.1 presents the data acquisition and road modeling methods used in previous representative studies and also describes the extent to which they meet the three road map requirements for intelligent vehicle systems. The vertical axis of the table represents the classification of the data acquisition methods based on the level of accuracy that the methods can meet, and the horizontal axis represents the classification of the road models based on the level of usability. The asterisk in

Table 2.1 Comparison to representative previous road map generation methods.

			Usability				
			N/A	Low	Mid.		High
		Road model / Data Acquis.	None	Polygon	Clothoid	B-spline	Set of piecewise polynomials
Accuracy	Meter	Aerial/satellite image		OSM[a] [17]			
	Decimeter	GPS+DR[b] (INS[c])			Bétaille, *et al.* [12]*	Jo, *et al.* [14]* .	
	Centimeter	GPS+DR (INS)+Camera			Ziegler, *et al.* [18]		
		GPS+DR (INS)+3D Lidar	Joshi, *et al.* [19]				Proposed*

*Methods where the maximization of storage efficiency is considered.

[a]Open street map, [b]Dead reckoning, [c]Inertial navigation system.

the table indicates that the marked method considers the maximization of storage efficiency. The table shows that no prior system has simultaneously achieved the three road map requirements for intelligent vehicle applications. A more detailed description about the previous approaches is presented in the rest of this section.

2.1.1 Acquisition of Road Geometry

Various approaches have attempted to acquire accurate road geometry data. For example, an aerial/satellite image-based approach has been extensively used for conventional digital maps and GISs. High-resolution aerial camera images have been acquired from a satellite or an aerial vehicle, to extract road geometries through manual work or through image processing means [20–22]. An image-based approach has an advantage in that we can obtain the road geometry for a larger region by processing a single image. However, the resolution of the aerial/satellite images is insufficient for extracting precise lane-level road geometry. Moreover, the elevation information of the roads cannot be acquired because the images contain no depth information. As a consequence, the accuracy of the image-based approach is limited to meter-level.

Many studies have adopted a probe vehicle-based approach to acquire a more accurate road geometry [11, 12, 14, 23–25]. In this approach, a probe vehicle equipped with various sensors explores roads and collects sensor data to obtain road geometry information. Of the various sensor configurations that are possible, kinematic GPS-based methods are the most widely used systems that include real-time kinematic (RTK) and post-processing kinematic (PPK) GPSs. In this method, the trajectory of a probe-vehicle driving along the centerline of a road (or lane) is recorded as road geometry data. The accuracy of the road geometry depends mainly on the posi-

tioning accuracy of the vehicle, so many previous studies have used algorithms that integrate GPS with other positioning systems, such as dead reckoning (DR) and inertial navigation system (INS), to improve the accuracy and reliability of the vehicle position [14, 26, 27].

Although the GPS-based methods are useful to obtain a greater degree of accuracy in the road geometry than image-based methods, there are some fundamental limitations to using this method. First, it is inefficient for the probe vehicle to capture information for a road multiple by as many times as the number of lanes. Such repetitive work is costly and time-consuming. The second problem is the so-called trajectory error. In a GPS-based method, it is desired for a probe vehicle to drive along the exact centerline of the lane. However, it is practically impossible for a human driver to control the vehicle with centimeter-level precision for an extended period of time. Therefore, discordance between the vehicle trajectory and the centerline of the lane is unavoidable. Thus the accuracy of using this method is limited to decimeter-level.

Various methods based on perception sensor have been introduced in recent years to tackle the problems of using GPS-based methods [18, 24, 28–31]. In these methods, road markings are detected and extracted by using perception sensors that capture road geometry data. Since the road geometry data is acquired directly from lane marking information, the above problems are fundamentally prevented. Various sensor configurations have been used to this end, including a monocular camera [24], stereo-camera [18], and 2D or 3D Lidar [19, 29–31]. Camera-based methods have an advantage in that the cost of the sensor can be reduced; however, it is difficult to extract precise 3D road geometry because of the inherent limitation of a camera, which is that the information is represented in a 2D plane. For this reason, the 3D Lidar is the most appropriate way to acquire centimeter-level road geometry since it

can provide accurate 3D information about roads. Some studies have extracted road geometry from 3D Lidar data. In [29], an algorithm was proposed to extract road geometry from 3D Lidar data; however, the algorithm focused on extracting road regions rather than lane-level road geometry. In [30] and [31], the extraction of lane markings from 3D Lidar data was considered, but, the clustering among lanes was not addressed. Joshi *et al.* proposed a particle filter-based method to extract lane-level geometry data and clustering among the lanes using 3D Lidar data [19]. However, that method extracts the centerline of lanes rather than the lane marking points, so the shape of the lane markings (e.g., the dashed line) is ignored, which can be useful for intelligent vehicle applications such as vehicle localization. In this chapter, I propose a 3D Lidar data processing algorithm to extract and cluster the lane marking points.

2.1.2 Modeling of Road Geometry

It is important to have the appropriate road geometry representation to ensure storage efficiency and usability along with map accuracy. Various road geometry models have been previously proposed [11, 12, 14, 24, 32, 33], but the previous models have not considered the three requirements simultaneously. For example, polygons are widely used in conventional digital road maps and various intelligent vehicle applications to represent the road geometry due to their simplicity [17, 34–37]. However, since a polygon cannot express curved roads with precision, too many line segments become necessary to express the roads with a large curvature in order to achieve centimeter-level accuracy. Therefore, the accuracy and storage efficiency requirements cannot be simultaneously satisfied. In addition, direct exaction of road geometry information cannot be provided, including the tangent angle and curvature, and thus the usability

requirement is also not satisfied.

Various mathematical curve models have been proposed to provide a more accurate and efficient road representation. For example, a clothoid is the best way to represent road geometry since a road is traditionally designed by a set of clothoids [12,38]. Bétaille *et al.* proposed a clothoid-based road modeling algorithm that satisfies the accuracy and storage efficiency requirements [12]. In this algorithm, the road geometry is represented as a set of the minimum number of clothoid curves while also satisfying a preset accuracy constraint. However, a clothoid is inadequate for use with intelligent vehicle applications since it includes transcendental functions. Thus the calculations for the information required for advanced vehicle applications, such as distance and relative angle between an ego-vehicle and a lane in the map, are computationally intense. As a result, the clothoid does not satisfy the usability requirement. Moreover, a clothoid cannot express the 3D road geometry by itself since the clothoid is only defined on a plane.

An alternative is to use a spline curve. Many studies have used various types of spline curves to represent the road geometry [11, 14, 24, 32, 33]. For example, a cubic B-spline is a representative method for road modeling that has an advantage in that local modifications of the curve do not affect to the entire shape of the curve. This is an important characteristic for road map maintainability, and in addition, many efficient B-spline curve approximation algorithms that have been developed for computer aided design (CAD) make it easy to use B-spline curves for road modeling. For example, in literature [14], a B-spline-based road modeling algorithm was proposed based on a dominant points-based B-spline curve fitting algorithm developed for computer-aided design (CAD) [39]. The control point of the B-spline is sub-optimally determined according to a preset accuracy constraint in order to simultaneously adhere to the

accuracy and storage-efficiency requirements. However, it is difficult to extract the information on the geometry of the road, such as the tangent angle and curvature by using a B-spline curve because the functions of the B-spline are not intuitive, and it is thus difficult to calculate the first and second derivatives of the functions [40]. Therefore, B-splines also do not satisfy the usability constraint.

In this dissertation, I use the most intuitive spline curve form where a curve segment is expressed as a piecewise cubic polynomial. Since a polynomial can be used to conveniently calculate the derivatives, we can extract the tangent angle and curvature of the road from the proposed spline curve by using simple arithmetic operations. Piecewise polynomial-based curve approximation has not been studied as much as other spline curves such as the B-spline. Moreover, previous piecewise polynomial approximation algorithms are not appropriate for approximating significantly large amount of road geometry data since the previous algorithms have exponential or $O(n^3)$ computational complexity [15, 16]. One solution is to approximate the data as a B-spline curve first and to convert the B-spine curve to a set of piecewise polynomials [41]. However, this approach requires a great number of piecewise polynomials than directly approximating the data as a set of piecewise polynomials as will be shown in Section 2.5.3. Thus, this chapter proposes an efficient piecewise polynomial curve approximation algorithm that processes a significantly large amount of road geometry data and sub-optimally determines the number of piecewise polynomials and their coefficients with $O(n)$ computational complexity.

2.2 Overall System Architecture

The overall road map generation system is composed of three subsystems: a data acquisition, a data processing and a road modeling subsystems (see Fig. 2.1). The data

acquisition and processing subsystems acquire accurate and reliable road geometry data by using various sensors during collection on a probe vehicle equipped with a GPS+INS vehicle positioning system and a 3D laser scanner. Then, accurate road geometry data is extracted by integrating the sensor data in the data processing system.

In this dissertation, two different types of road geometry are acquired depending on the road type. For a road in which lane markings exist, the Cartesian coordinates of the lane markings are obtained as the road geometry data from the 3D laser scanning data (see the black circles in Fig. 2.1). For a road in which lane markings do not exist, such as when going off-road, the trajectory of a probe vehicle driving along centerline of a road is used in a manner similar to the previous probe vehicle-based approach (see the white squares in Fig. 2.1). For convenience, I have name the road in which lane markings exist as a *type I* road and the other as a *type II* road in the rest of this chapter.

Both types of road geometry data obtained from the data acquisition and processing systems are represented by a large number of points. However, this type of geometry representation is inadequate for a road map in terms of the storage efficiency. Moreover, the point representation makes it difficult to extract the required road geometry information, such as tangent angle and the curvature of a road. To solve these problems, I have applied a mathematical modeling technique to the road geometry point data in order to increase its storage efficiency and usability. The road modeling system approximates a set of geometry point data that correspond to a road line as a cubic spline curve consisting of sequential parametric piecewise cubic polynomials curves (see the right bottom of Fig. 2.1). A *sequential approximation algorithm* is proposed to efficiently approximate the curve.

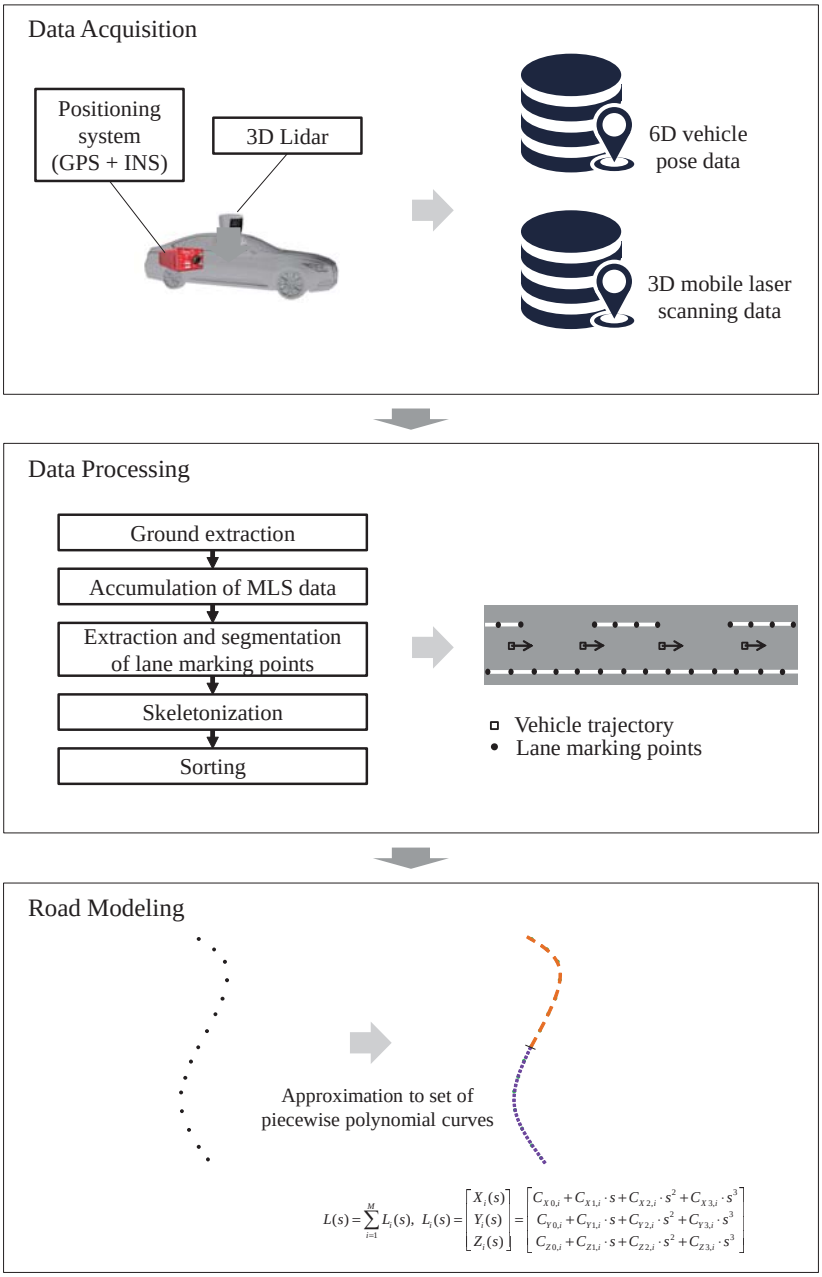


Figure 2.1 Overall System Architecture.

2.3 Road Geometry Data Acquisition and Processing

In the data acquisition, the required raw sensor data is collected by using a probe vehicle equipped with a positioning system and a 3D Lidar, and the geometry data

of the lanes is then obtained from raw data in the data processing step. Detailed description of the two steps is presented as follows.

2.3.1 Data Acquisition

During the data acquisition step, two kinds of data are collected by the probe vehicle: 6D vehicle pose data and 3D MLS data. The vehicle pose data includes the accurate 3D global position and 3D attitude (yaw, pitch and roll) with respect to the trajectory of the probe vehicle. In order to obtain the accurate vehicle pose data, a high-precision vehicle positioning system is recommended. In this dissertation, I use a sensor fusion system that integrates a RTK-GPS and a high-precision INS. The 3D MLS data is obtained by correcting the raw 3D laser scanning data using the INS data. When a vehicle moves, the measurement origin of the laser scanner is changed; thus the laser points are twisted [42]. To resolve this problem, a preprocessor untwists the points using the vehicle motion sensor measurements from the INS.

For *type II* roads, the vehicle pose data is simply used as the road geometry data, and for *type I* roads, MLS data is additionally used. The rest of this section presents the data processing algorithm that is used to obtain accurate road geometry data from the raw data. Note that this dissertation focuses more on a *type I* road since the road geometry for *type II* roads can be obtained by using the previous probe vehicle-based methods.

2.3.2 Data Processing

The road geometry data for a *type I* road can be obtained by integrating 3D MLS data with the vehicle pose data. Fig. 2.2 depicts the overall procedure for the data processing step. First, ground extraction is applied to each frame of the MLS data

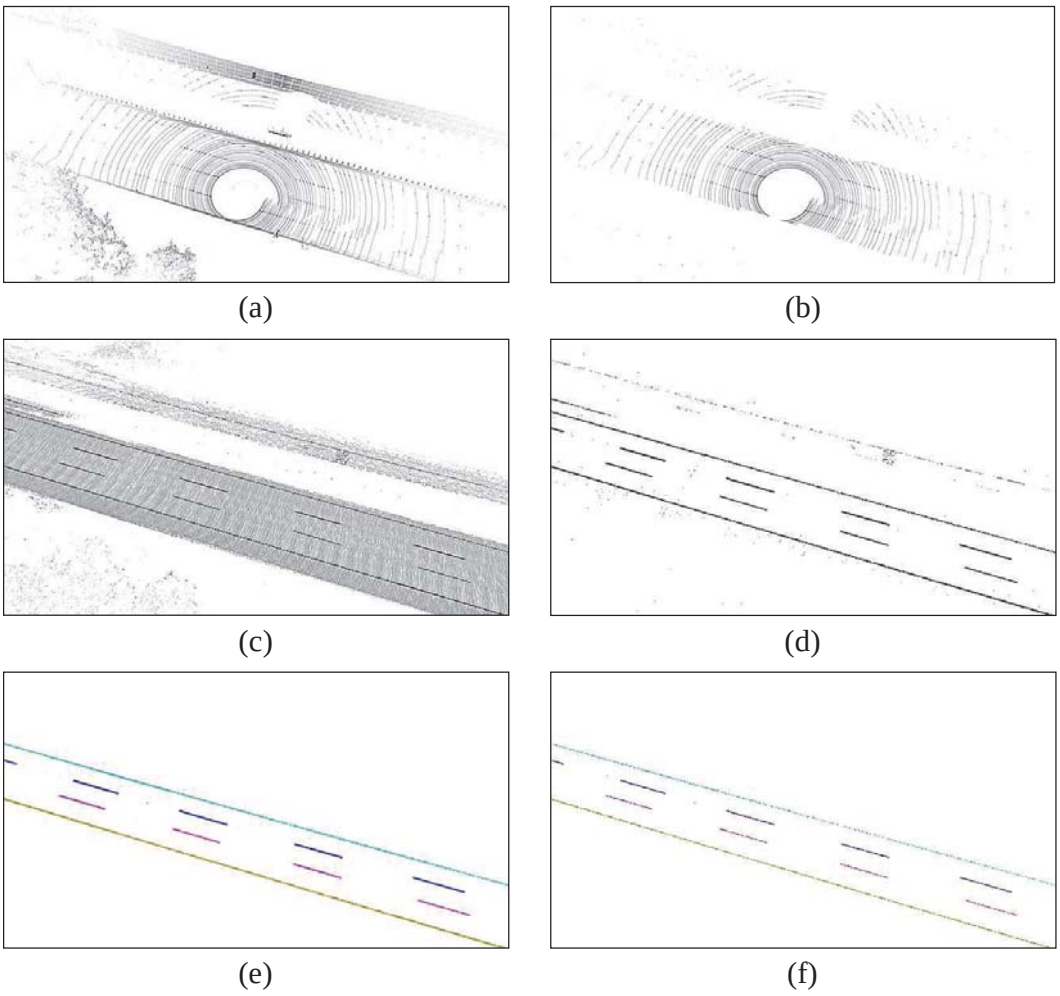


Figure 2.2 Overall data processing procedure: (a) raw MLS data, (b) ground extraction result, (c) result of data accumulation, (d) result of intensity thresholding, (e) result of lane marking point extraction and clustering, and (f) skeletonization result.

to remove unnecessary points from the MLS data, and then, the ground points for every frame of the MLS data are accumulated on a global coordinate system by using 6D vehicle pose data synchronized with MLS data. The accumulated ground points with a high intensity are then extracted as candidate points that correspond to lane markings, and the exact lane marking points are extracted and clustered by

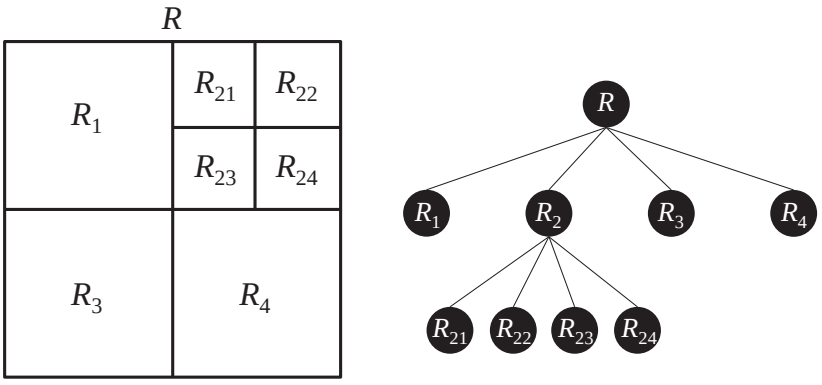


Figure 2.3 Quadtree partition structure which is used for ground extraction.

using more complex algorithms. Finally, the desired geometry data is obtained by skeletonizing the dense point data.

Ground extraction

Since the lane marking points on the ground surface are the only points of interest in this dissertation, removing off-ground points from each MLS is helpful for later processes in terms of the processing time and performance. Ground extraction is carried out according to an adaptive quadtree partition structure, as shown in Fig. 2.3. Initially, the MLS data points are divided into multiple cells with a size of w_{init}, and then, each cell is tested to identify whether the cell is a ground cell or not. If it is not certain whether a cell is a ground cell or not, the cell is divided into four child cells, and the test is applied to all child cells. This process is then continued until all the cells have been identified. The test to identify this type of cell is conducted as follows.

First I assume that there is at least one cell among the initial cells that has already been identified as a ground cell. This assumption is valid since it is certain that the

cells near the probe vehicle are ground cells. Then, the test is performed starting from the unidentified cells neighboring the identified cells. Let an unidentified cell be R_{UI}. Then the roughness of the cell is calculated to determine whether the cell is a ground cell or not by

$$\Delta z_{cell} = z_{\max}(R_{UI}) - z_{\min}(R_{UI}), \tag{2.1}$$

where $z_{\max}(\cdot)$ and $z_{\min}(\cdot)$ refer to the z coordinate value of the highest points and the lowest point in the cell, respectively. If the roughness value of the cell is larger than a given threshold, this means that the cell has to include the off-ground points. In this case, the cell is unidentified, and thus is divided into four child cells again. On the other hand, if the roughness value of the cell is smaller than the given threshold, it is then considered to be a candidate ground cell. For a candidate ground cell, the identification whether the cell is actually a ground cell is performed according to the following test:

$$|z_{\text{mean}}(R_{\text{UI}}) - z_{\text{mean}}(R_{\text{I,Nbh}})| < \lambda_{\text{GND}} \tag{2.2}$$

where $R_{\text{I,Nbh}}$ denotes the ground cell that has already been identified among the neighborhood of the test cell R_{UI}, and $z_{\text{mean}}(\cdot)$ refers to the mean value of heights of the points. If the test result is true, the height difference between the test cell and the neighbor ground cell is small, and therefore, the cell is identified as a ground cell. If the test result is false, then the test cell is identified as an off-ground cell. As a consequence, a cell is categorized into three states after this process: a ground cell, an off-ground cell or an unidentified cell. For the unidentified cells, the above process is repeated until all cells have been identified. Fig. 2.2b shows an example of the ground extraction.

Accumulation of MLS data

Every frame of the MLS data from which off-ground points have been removed is accumulated on a global coordinate system by using synchronized vehicle pose data. Basically, the MLS data is represented on a vehicle body coordinate system, and therefore, a rigid body transformation is applied to accumulate these in a global coordinate system. The rigid body transformation is simply defined by a transformation matrix, e.g.,

$$T_H = \begin{bmatrix} \cos\alpha\cos\beta & \cos\alpha\sin\beta\sin\gamma - \sin\alpha\cos\gamma & \cos\alpha\sin\beta\cos\gamma + \sin\alpha\sin\gamma & X_t \\ \sin\alpha\cos\beta & \sin\alpha\cos\beta\sin\gamma + \cos\alpha\cos\gamma & \sin\alpha\cos\beta\cos\gamma - \sin\alpha\sin\gamma & Y_t \\ -\sin\beta & \cos\beta\sin\gamma & \cos\beta\cos\gamma & Z_t \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{2.3}$$

where α, β and γ are the Euler angles of the vehicle body, respectively, and (X_t, Y_t, Z_t) represents the position of the vehicle in a global coordinate system. The MLS data is accumulated on the global coordinate system by the rigid body transformation, and finally, dense point cloud data representing the ground region is obtained, as shown in Fig. 2.2(c).

Extraction and Clustering of Lane Marking Points

Now, the points corresponding to the lane markings are extracted and are clustered among the points that belong to the same road line. First, rough lane marking points are extracted by using simple intensity-based thresholding, i.e., the points with a higher intensity than the threshold is extracted. I assume that an appropriate static intensity threshold is given for a road, since the intensity value of the Lidar is robust to external factors such as illumination. As we can see in Fig. 2.2(d), the majority of

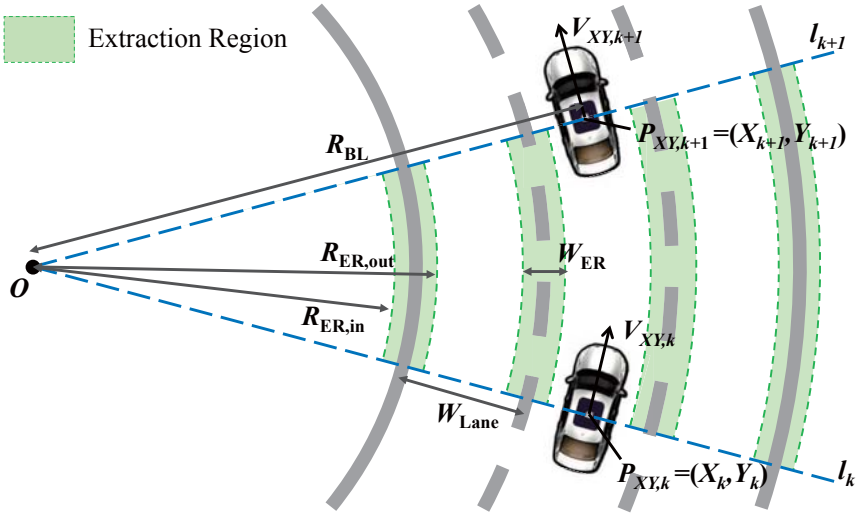


Figure 2.4 Lane marking point extraction and clustering.

non-lane marking points are effectively filtered out just through the simple intensity thresholding method. However, there are still exist outliers, and in addition, the points are not clustered by the road line. Therefore, an additional algorithm is applied to remove the remaining outliers and to cluster the points by the road line, and since it is difficult and inefficient to process all points at once, I first vertically divide the point cloud into a set of data blocks by using the vehicle pose data in the XY plane. A block is then defined by two lines, named a split line, where the kth split line, l_k, is a line that is orthogonal to the yaw of the kth vehicle pose and the passing vehicle position $P_{XY,k}$ (see the coarse dashed lines in Fig. 2.4). With the split lines, all points are divided into $N_p - 1$ blocks, where N_p is the number for the vehicle pose data (number of split lines). These data blocks are then processed separately to extract the lane marking points as follows.

The algorithm starts by finding an intersection for two split lines, l_k and l_{k+1}, and the radius R_{BL} with respect to the intersection point. Note that I assume that

the probe vehicle drives along the centerline of a lane as much as possible, and the vehicle trajectory between time k and $k+1$ can be modeled by an arc since the time gap is small enough (10 ms in this dissertation). After that, the lane marking point extraction regions in which the lane marking points are likely to exist are set using some prior information of the road, e.g, the number of lanes, N_{L}; width of a lane, W_{L}; and the lane number in which the probe vehicle drove, n_{drv} ($n_{\mathrm{drv}} = 1$ for the leftmost lane). This information can be obtained using conventional maps, such as Open Street Map (OSM) [17], or by using camera-based algorithms [43–45]. There is an exceptional case where the yaw angles of two consecutive vehicle pose data are exactly the same; thus, the intersection point of two split lines does not exist. In this case, the radius R_{BL} is set to a value that is large enough (e.g., 3000 m), and the origin O is set to a point so that it satisfies $\|P_{XY,k} - O\| = \|P_{XY,k+1} - O\| = R_{\mathrm{BL}}$. This trick is valid since the radius is large enough compared to the length of the arc between points $P_{XY,k}$ and $P_{XY,k+1}$ (about 2.7 m where the speed of the vehicle is 100 km/h and the pose data is acquired at a 10 Hz data rate) As shown in Fig. 2.4, the extraction region is bounded by two inner and outer arcs, and the radiuses of the arcs are determined by R_{BL}, N_{L} and W_{L} as follows:

$$R_{\mathrm{ER},i} = \{R_{\mathrm{ER},i,\mathrm{in}}, R_{\mathrm{ER},i,\mathrm{out}}\} = R_{\mathrm{BL}} + (2i - 2n_{\mathrm{drv}} - 1)\frac{W_L}{2} \pm \frac{W_{ER}}{2}, \tag{2.4}$$

where i denotes the index of the road lines from the left (e.g., $i = 1$ for the leftmost road line), and the width of the extraction region, W_{ER}, is determined by considering the uncertainty degree with respect to the error between the expected location of the lane marking points and the true location of these. Although the larger value for W_{ER} may decrease the false negative rate, more non-lane marking points may be extracted. On the other hand, a smaller W_{ER} value may exclude non-lane marking points as well, which would decrease the true positive rate. Therefore, it is important to carefully

determine W_{ER} value to ensure reliable road marking point extraction. Finally, the points inside of the extraction region are extracted as the lane marking points, and the points belonging to the same region are grouped together. After finishing the extraction procedure, the points belonging to the same region are naturally grouped, and thus the lane marking points are clustered into several clusters that correspond to each of the road lines.

Skeletonization

In the previous steps, the lane marking points were obtained as shown in Fig. 2.2(e). However, since the lane marking points describe thick lane markings at this stage, it is difficult to represent lane markings as lines by using this set of lane marking points. Skeletonization is applied to the lane marking points to represent the lane markings as lines, as shown in Fig. 2.2(f). Skeletonization is a technique that is used to simplify and abstract a volumetric object to a line shape [46]. In this dissertation, the skeletonization is applied to lane marking points for extracting the points that correspond to the centerlines of the lane markings.

Sorting

For the final process, the lane marking points within each cluster are sorted and sequentially indexed from the closest point to the farthest point with respect to the starting point of the road. However, applying the sorting algorithm to all of the points is time-consuming. Fortunately, the points were divided into multiple blocks in the lane marking point extraction process. Since the blocks are sequentially arranged from the starting point of the road, we can sort all of the points by sorting the points in each block separately and concatenating the blocks of sorted points. In this dissertation,

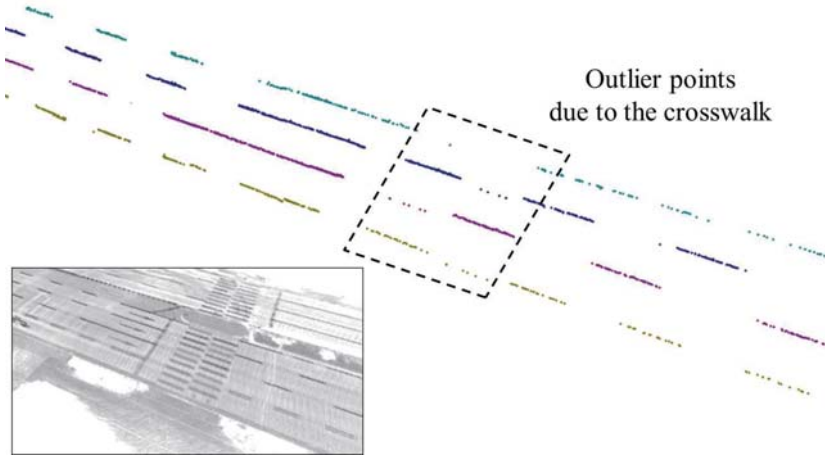


Figure 2.5 Example of outlier points.

the bubble sorting algorithm is used in a block. As a result of the data sorting, the geometry data of the road with respect to the road lines is represented as sets of sequential points as follows:

$$\begin{aligned} \mathcal{G} &= \{\mathcal{G}_i | i = 1, ..., N_L + 1\} \\ \mathcal{G}_i &= \{G_n = (X, Y, Z)_n | n = 1, ..., N_i\}, \end{aligned} \tag{2.5}$$

where $\mathcal{G}_i$ is a set of sequential lane marking points corresponding to the ith road line, N_L is the number of lanes, G_n is the nth lane marking point, and N_i is the total number of lane marking points in $\mathcal{G}_i$.

2.3.3 Outlier Problem

The MLS-based lane marking extraction algorithm is susceptible to outliers that can be extracted with true lane marking points. As shown in Fig. 2.5, the road markings that exist outside of the extraction region such as arrows are effectively excluded by the extraction algorithm. However, the road markings that exist inside the extraction region (road markings that are well aligned with the lane markings) are extracted

with the lane markings. These outliers may prevent the extracted road geometry data points from accurately expressing the true geometry of the road. This problem is supplemented by the road modeling algorithm that will be proposed in a later section. The proposed road modeling algorithm was designed to accurately model the true geometry of the road even though there exist outliers in the data.

2.4 Road Modeling

The road geometry data $\mathcal{G}$ that is represented by a set of points is not suitable for use as a road map since it consumes a large amount of storage space and the road geometry information is difficult to extract from the data. In this section, I propose a road modeling algorithm that increases the storage efficiency and usability of a road map by representing each set of points, $\mathcal{G}_i$, corresponding to a road line, as a mathematical curve defined as

$$L(s) = \sum_{m=1}^{M} L_m(s), \tag{2.6}$$

where, the subscript i that refers to the index of the road line was omitted for convenience, and

$$L_m(s) = \begin{bmatrix} X_m(s) \\ Y_m(s) \\ Z_m(s) \end{bmatrix} = \begin{bmatrix} \sum_{p=0}^{3} C_{Xp,m} \cdot (s - s_m)^p \\ \sum_{p=0}^{3} C_{Yp,m} \cdot (s - s_m)^p \\ \sum_{p=0}^{3} C_{Zp,m} \cdot (s - s_m)^p \end{bmatrix}, \; for\, s \in [s_m, s_{m+1}). \tag{2.7}$$

The curve $L(s)$ is a cubic spline curve that is composed of a finite number of sequentially connected piecewise polynomial curves. Each piecewise polynomial curve is parameterized according to $s \in [s_i, s_{i+1}]$ and consists of three cubic polynomials

that represent the X, Y and Z coordinates, respectively. The advantage of representing the road geometry data points as a spline curve is obvious since the number points can be expressed as only one parameter value and twelve coefficients of a piecewise polynomial, i.e., s_m, $C_{X0,m} \sim C_{X3,m}$, $C_{Y0,m} \sim C_{Y3,m}$ and $C_{Z0,m} \sim C_{Z3,m}$.

The problem lies in expressing the point data in the spline curve form, that is, how to determine the number of piecewise polynomial curves and coefficients of the curves. There are three main considerations for this problem: storage efficiency, accuracy and outlier handling. First, the number of piecewise polynomial curves that compose the spline curve should be minimized to maximize the storage efficiency. However, there is a tradeoff between the number of piecewise polynomials and the accuracy, and in general, as we use a greater number of polynomial curves, the spline curves can express the point data with higher accuracy. Therefore, the problem is to minimize the number of piecewise polynomial curves while satisfying predefined accuracy constraint. Meanwhile, outliers included in the point data negatively impact the accuracy of the curve approximation. Therefore, the goal of the road modeling system is to find the optimal M, the number of piecewise polynomial curves, $\{s_m | m = 1, ..., M\}$, the set of parameters that indicate the point where the piecewise polynomial curves are divided, and the coefficients of the curves that take the accuracy constraint and outlier points into account.

However, it is very difficult to solve this problem from a global optimization point of view since we need to consider the number of piecewise polynomials, the parameter points at which the curve is divided, and the coefficients of the polynomials altogether. In addition, the accuracy constraint and the outliers should also be considered. Therefore, I consider the original problem as a combination of smaller problems that can be solved relatively easily, and I thus propose the *sequential approximation algorithm*

to efficiently find the near-optimal solution. In the rest of this section, I describe the details of the sequential approximation algorithm.

2.4.1 Overview of the sequential approximation algorithm

As mentioned above, the basic idea to solve the global optimization problem is to convert the problem into a combination of multiple simpler problems. The problem conversion process is based on the intuition that minimizing the number of piecewise polynomials that model the overall road geometry data is equivalent to maximizing the number of data points that each piecewise polynomial can express. I thus propose the sequential approximation algorithm, which can be summarized as below:

i. A set of road geometry point, $\mathcal{G} = \{G_n = (X, Y, Z)_n | n = 1, ..., N\}$, is given. Note that index i of the point set $\mathcal{G}_i$ is omitted for convenience.

ii. An initial polynomial curve is approximated by using the first few points. Here, the starting point of the curve is fixed for the first data point.

iii. Starting from the second point, a new point is sequentially added at every step, and the coefficients of the polynomial curve are corrected by using a Kalman filter (KF)-based algorithm.

iv. Whenever the polynomial curve is corrected, the accuracy of the curve is verified.

v. If the accuracy is higher than a threshold, step iii) is repeated. If not, adding new data points stops, and the coefficients of the polynomial curve that do not violate the accuracy criterion are saved.

vi. The remaining data points are used to repeat, ii) - v) until the last data point is has been reached.

As can be seen from the above description of the algorithm, the overall algorithm is composed of multiple sequential approximation processes. In an approximation process, a piecewise polynomial curve is generated to expresses as many points as possible within the accuracy bound. This approach is used to sub-optimally determine the set of break parameters, $\{s_m | i = m, ..., M\}$, and the coefficients of the curves. In the rest of this section, a detailed explanation of the sequential approximation algorithm is presented. First, I describe the detailed curve approximation process, and then present the detailed curve transition criterion and algorithm. The overall procedure for the algorithm is described in Algorithm 1.

2.4.2 Approximation Process

I use a Kalman filter, a recursive Bayesian estimation approach to approximate the given points to a cubic polynomial curve. The data points are regarded as observations for the estimation system, and the coefficients of the polynomial curve are regarded as a system state that we want to estimate. The reason to use a stochastic method instead of deterministic methods, such as spline interpolations, is that the data points contain stochastic errors even though the data has been refined in the data acquisition and processing systems. In addition, outliers included in the data should also be handled, and with proper parameter settings, a Bayesian filter can be designed to be robust against such outliers.

System Model

The system state vector x_k and the measurement vector y_k are defined by

$$\begin{aligned} x_k &= (C_{X1,k}, C_{X2,k}, C_{X3,k}, C_{Y1,k}, C_{Y2,k}, C_{Y3,k}, C_{Z1,k}, C_{Z2,k}, C_{Z3,k})^{\mathrm{T}}, \\ y_k &= (X_{y,k}, Y_{y,k}, Z_{y,k})^{\mathrm{T}}. \end{aligned} \tag{2.8}$$

Algorithm 1: Sequential Approximation Algorithm

input : $\mathcal{G}_i = \{G_{i,n} = \{X, Y, Z\}_{i,n} | n = 1, ..., N_i\}$

output: $L_i = \{L_{i,m} | m = 1, ..., M_i\}$

$m \leftarrow 1$; $k \leftarrow 1$; $s_k \leftarrow 0$

$[C_{0,m},\, x_{m,k|k},\, P_{m,k|k}] \leftarrow$ `Initialization`$(\mathcal{G}_i,\, k)$;

$[x_{m,k+1|k},\, P_{m,k+1|k}] \leftarrow$ `Prediction`$(x_{m,k|k},\, P_{m,k|k})$;

for $k = 2$ **to** N_i **do**

  $s_k \leftarrow s_{k-1}+$`DistanceBetweenPoints`$(G_{i,k-1},\, G_{i,k})$;

  if `OutlierCheck`$(x_{m,k+1|k},\, P_{m,k+1|k},\, G_{i,k})$ *false* **then**

    $[x_{m,k|k}, P_{m,k|k}] \leftarrow$ `Update`$(x_{m,k+1|k},\, P_{m,k+1|k},\, u_k,\, \mathcal{G}_{i,k})$;

  else

    $x_{m,k|k} = x_{m,k|k-1}$;

    $P_{m,k|k} = P_{m,k|k-1}$;

  end

  if (`CurveTransitionTrigger`$(\mathcal{G}_i,\, C_{0,m},\, x_{m,k|k})$) or $(k = N_i)$ **then**

    $k \leftarrow k - N_{\text{buff}}$;

    $L_{i,m} \leftarrow \{C_{0,m},\, x_{m,k|k}\}$;

    $m \leftarrow m + 1$;

    $[C_{0,m},\, x_{m,k|k},\, P_{m,k|k}] \leftarrow$ `Initialization`$(\mathcal{G}_i,\, k)$;

  end

  $[x_{m,k+1|k},\, P_{m,k+1|k}] \leftarrow$ `Prediction`$(x_{m,k|k},\, P_{m,k|k})$;

end

$L_i \leftarrow$ `ArcLengthParameterization`(L_i);

/*$C_{0,m} = \{C_{X0,m}, C_{Y0,m}, C_{Z0,m}\}$ */

Note that the constant terms of the polynomials are excluded from the system state vector to ensure the geometrical continuity between neighbouring piecewise curves. However, I do not regulate both C_1- and C_2-*continuity* since there can be points where the tangent or the curvature are discontinuous in real roads.

Based on the state and the measurement vector definitions, I establish linear prediction and measurement models that are suitable for Kalman filtering. The prediction model is defined as

$$\begin{aligned} & x_k = x_{k-1} + w_{k-1},\ w_{k-1} \sim N(0, Q_{k-1}) \\ \Rightarrow & \begin{bmatrix} \mathbf{C}_{X,k} \\ \mathbf{C}_{Y,k} \\ \mathbf{C}_{Z,k} \end{bmatrix} = \begin{bmatrix} \mathbf{C}_{X,k-1} \\ \mathbf{C}_{Y,k-1} \\ \mathbf{C}_{Z,k-1} \end{bmatrix} + w_{k-1}, \end{aligned} \tag{2.9}$$

and the measurement model is defined as

$$\begin{aligned} y_k &= H_k \cdot x_k + u_k + v_k,\ v_k \sim N(0, R_k) \\ y_k &= \begin{bmatrix} X_{\mathrm{y},k} \\ Y_{\mathrm{y},k} \\ Z_{\mathrm{y},k} \end{bmatrix} = \begin{bmatrix} C_{X0} + C_{X1,k} \cdot s_k + C_{X2,k} \cdot s_k^2 + C_{X3,k} \cdot s_k^3 \\ C_{Y0} + C_{Y1,k} \cdot s_k + C_{Y2,k} \cdot s_k^2 + C_{Y3,k} \cdot s_k^3 \\ C_{Z0} + C_{Y1,k} \cdot s_k + C_{Y2,k} \cdot s_k^2 + C_{Y3,k} \cdot s_k^3 \end{bmatrix} + v_k \\ &= \begin{bmatrix} s_k & s_k^2 & s_k^3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & s_k & s_k^2 & s_k^3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & s_k & s_k^2 & s_k^3 \end{bmatrix} \cdot x_k + \begin{bmatrix} C_{X0} \\ C_{Y0} \\ C_{Z0} \end{bmatrix} + v_k, \end{aligned} \tag{2.10}$$

where k denotes the iteration step of the Kalman filter, s_k is a system input denoting the parameter value of the polynomial curve, and w_k and v_k are the prediction and measurement noises with covariance Q_k and R_k, respectively. Under ideal conditions where the prediction noise is ignored, the system state does not change during the prediction step. This is reasonable since the coefficients of the polynomial curve are

only affected by the data points that are observed. The coefficients should be adjusted only when a new data point has been added.

Measurement matrix H_k is a function of the time-varying system input s_k. Ideally, s_k is an arc-length parameter of the spline curve that corresponds to the measurement point y_k. However, since it is impossible to calculate the arc-length of the curve before finishing the approximation process, I approximate the curve by using arbitrary, strictly-increasing parameters first, and then I re-parameterize the curve by the arc-length after finishing the approximation process. To simplify the re-parameterization process, I use the chord-length from the initial point as the parameter value, s_k, which is an approximated value of the arc-length.

Observability Proof

The suitability of the proposed system model can be proven by checking the observability of the system. Observability is a necessary condition for the Kalman filter to work correctly, and we say that a linear system is observable if there is a finite number of steps n so that knowledge about the input sequence $u_0, ..., u_{n-1}$ and the output sequence $y_0, ..., y_{n-1}$ is sufficient to determine the initial state of the system, x_0. Therefore, the system is observable if x_0 is uniquely determined by

$$\begin{bmatrix} y_0 \\ \vdots \\ y_{n-1} \end{bmatrix} = O_n x_0 + T_n \begin{bmatrix} u_0 \\ \vdots \\ u_{n-1} \end{bmatrix}, \tag{2.11}$$

where O_n is a matrix that maps the initial state x_0 into the resulting output over $[0, n-1]$ and T_n is a matrix that maps the input to the output over $[0, n-1]$. For

our system, the above problem can be represented as

$$\begin{bmatrix} y_0 \\ \vdots \\ y_{n-1} \end{bmatrix} = \begin{bmatrix} H_0 \\ \vdots \\ H_{n-1} \end{bmatrix} x_0 + \begin{bmatrix} C_0 \\ \vdots \\ C_0 \end{bmatrix}, \tag{2.12}$$

where $C_0 = [C_{X0}, C_{Y0}, C_{Z0}]^{\mathrm{T}}$. For $n = 3$, the initial state $x(0)$ is uniquely determined if and only if $N(O_3) = 0$ and, equivalently, $\mathrm{Rank}(O_3) = 9$. The matrix O_3 is composed of three observation matrices: H_0, H_1 and H_2. It can be easily proved that $\mathrm{Rank}(O_n) = 9$. By definition, since $s_k > s_{k-1}$ for every k, every row vector for matrix O_n is independent relative to the other row vectors. Therefore, I can prove that the proposed system is observable.

Error Covariance

In order to obtain good results from the Kalman filter, the proper error covariance matrices, Q_k and R_k, need to be set up. First, the prediction error should be zero (and thus $Q_k = \mathbf{0}$) for this problem since the statistics of the state values do not vary during the prediction step.

The measurement error covariance matrix, R_k, is determined according to the statistics of the vehicle pose data and the MLS data. For example,

$$R_k = \begin{bmatrix} 0.05^2 & 0 & 0 \\ 0 & 0.05^2 & 0 \\ 0 & 0 & 0.05^2 \end{bmatrix}. \tag{2.13}$$

Processing Outlier Points

The Kalman filter has the inherent ability to handle outliers by allocating small gains to measurements that are far from the predicted system state. However, outliers that

are too large or too frequent may force the Kalman filter to diverge. To address this problem, outliers are detected and isolated so that they are not used for state update, by applying a normalized innovation squared (NIS) test to every measurement point before updating the system state using the measurement point. The NIS test is defined as follows:

$$\text{NIS}_k = (y_k - H_k\hat{x}_{k|k-1})^T S_k^{-1}(y_k - H_k\hat{x}_{k|k-1}) > \lambda, \tag{2.14}$$

where

$$S_k = H_k P_{k|k-1} H_k^T + R_k. \tag{2.15}$$

The measurement point that is larger than a threshold λ is considered to be an outlier, and it is discarded. The threshold value λ should be carefully determined since a value that is too large for λ negatively impacts the ability to detect for an outlier, while the Kalman filter may diverge with a too small value for λ due to frequent removal of normal measurement points.

2.4.3 Curve Transition

Based on the proposed system and measurement model, covariance matrices, and outlier point process algorithm, the KF approximates the road geometry point data that are given one by one to a polynomial curve. At a certain point, the KF should stop the current approximation and should start to generate a new polynomial curve since one cubic polynomial curve cannot express all points. The condition and procedure for the curve transition are discussed below.

At the end of every step of the KF, two kinds of maximum distance between the curve and the data points are calculated in order to decide whether the new curve

approximation should be started, i.e.,

$$\begin{aligned} D_{\max,XY} &= \max_i \left\| L_{XY}(\bar{s}_i) - G_{XY,i} \right\|, \; i = 1, ..., k, \\ D_{\max,Z} &= \max_i \left\| L_Z(\bar{s}_i) - G_{Z,i} \right\|, \; i = 1, ...k, \end{aligned} \tag{2.16}$$

where

$$L_{XY}(s) = \begin{bmatrix} C_{X0} + C_{X1} \cdot s + C_{X2} \cdot s^2 + C_{X3} \cdot s^3 \\ C_{Y0} + C_{Y1} \cdot s + C_{Y2} \cdot s^2 + C_{Y3} \cdot s^3 \end{bmatrix} \tag{2.17}$$

and

$$L_Z(s) = C_{Z0} + C_{Z1} \cdot s + C_{Z2} \cdot s^2 + C_{Z3} \cdot s^3 \tag{2.18}$$

are the polynomial curve on the XY- and Z-planes, respectively, while $G_{XY,i}$ and $G_{Z,i}$ refer to the i-th data point on the XY- and Z-plane, respectively. Therefore, $D_{\max,XY}$ and $D_{\max,Z}$ refer to the maximum distance errors between the curve and the data points on the XY- and Z-plane, respectively. The reason to calculate the distance error on the XY- and Z-plane separately is that the accuracy requirements for the XY- and Z-plane can be different. In general, autonomous vehicle systems require accuracy at the centimeter-level in the XY-plane since the map accuracy in the XY-plane affects vehicle safety in applications that involve vehicle localization or vehicle motion planning. On the other hand, a relatively low accuracy is required for the Z-plane since generally the height information of the road is generally used in non-safety-related applications such as for fuel management systems.

Ideally, the curve transition should be conducted at the moment when at least one of the two distance error exceeds a pre-defined threshold, i.e.,

$$D_{\max,XY} > \chi_{XY} \text{ or } D_{\max,Z} > \chi_Z, \tag{2.19}$$

However, the maximum distance error can occasionally exceed the threshold even though all data points belong to one polynomial curve due to stochastic errors that

are included in the data. In this case, the errors have to be ignored, and the KF should be continued. For this reason, a heuristic test is applied for the curve transition as follows:

i. A counter is initialized to zero before starting the KF.

ii. At the end of the every step of the KF, the maximum distance errors, $D_{\max,XY}$ and $D_{\max,Z}$, are calculated.

iii. The counter increases if $D_{\max,XY} > \chi_{XY}$ or $D_{\max,\mathrm{Z}} > \chi_Z$ and it is set to zero otherwise.

iv. If the counter exceeds a threshold N_{buff}, the curve transition is triggered.

In summary, the curve transition is triggered when the maximum distance errors exceed the threshold N_{err} times in a row. Once the curve transition is triggered at step k, the curve approximated at step $k-N_{\text{buff}}$ is saved since the curve approximated from step $k-N_{\text{buff}}$ to k is inaccurate. Therefore, state vectors and covariance matrices for the KF for recent N_{buff} steps should be stored in the buffer, and after the current polynomial curve is saved, the KF is initialized and a new curve approximation process begins.

In addition to the normal case, there is another case where the curve transition is needed. In a road, the number of lanes can vary, as shown in Fig. 2.7. In this case, the curve approximation process has to be stopped at the point where the road line disappears, and a new curve has to be started from the point where the road line reappears. In order to handle this case, at every step of the approximation process, the distance between the current data point and the next point is calculated. If the distance is larger than a threshold, i.e.,

$$\|G_{i+1} - G_i\| > \eta, \tag{2.20}$$

which means that the road line disappears at the current point, the curve transition is triggered. Since the new curve is discontinuous with the previous curve, the curve parameter representing the arc-length is set to zero at the start point of the new curve.

2.4.4 Arc length parameterization

The remaining portion of the procedure for road modeling involves arc-length parameterization. As mentioned in the previous sections, the curve generated from the sequential approximation algorithm is parameterized according to the chord length as an approximation of the arc-length. However, since accurate arc-length information is useful for many intelligent vehicle applications, I parameterize the curve in terms of the arc-length as the final task for the road modeling. Many studies have investigated the arc-length parameterization task for a curve, and in this dissertation, I use the algorithm proposed in [47] since it can process a large quantity of map data in a simple and efficient manner. The arc-length parameterization of a curve can be constructed by adhering to the following two-step process:

i. Compute the arc length s as a function of parameter $t : s = A(t)$. Since s is a strictly increasing function of t, there is an one-to-one correspondence between s and t.

ii. $t = A^{-1}(s)$, inverse of the arc-length function. This function is well defined and monotonically increases for cubic splines. By substituting $t = A^{-1}(s)$ into $Q(t)$, I obtain a curve that is parameterized by the arc length s, $P(s) = (X(A^{-1}(s), Y(A^{-1}(s), Z(A^{-1}(s)))))$, where $s \in [0, D]$ and D is the total length of the curve.

Figure 2.6 Probe vehicle equipped with a 3D Lidar and a GPS+INS positioning system.

Further details of the algorithm are provided in the literature [47]. After conducting the arc length parametrization, we obtain the final arc length-parameterized cubic spline curve for the data points of the road geometry.

2.5 Experimental Validation

In this section, I provide the results of the experiment that was carried out to evaluate the performance of the proposed road map generation system. First, I introduce the probe vehicle and sensor configuration used in the experiments. I then present the results of 3D Lidar-based data acquisition and data processing. I also evaluate the accuracy, storage efficiency and usability performance of the road modeling system. Finally, I provide an autonomous driving test based on the proposed road map to validate the practicality of the proposed road map system.

2.5.1 Experimental Setup

The raw data required to evaluate the proposed road map generation system was collected using a probe vehicle equipped with a 3D Lidar (Velodyne HDL-64E) and

Table 2.2 Specification of the 3D Lidar.

Horizontal field of view	Vertical field of view	Distance accuracy	Measurement rate
360 deg.	26.8 deg.	<2 cm	>1.3 M points/sec.

Table 2.3 Specification of the vehicle positioning system.

Measurement	Frequency	Accuracy (RMS)
Position	100 Hz	2 cm
Yaw	100 Hz	0.1 deg.
Roll/pitch	100 Hz	0.03 deg.

a GPS+INS vehicle positioning system (OXTS RT3002) as shown in Fig. 2.6. The Velodyne HDL-64E is a 64-layer Lidar, and it was configured to rotate 360 degrees to emit 64-layer laser beams with a minimum of 1.3 million points measured per second. OXTS RT3002 is a unified GPS+INS system that provides a highly accurate 6D vehicle poses by combining the GPS and INS data. The specifications of these two measurement system are summarized in Tables 2.2 and 2.3, respectively. In this dissertation, raw data was collected with a 10 Hz rate, and the prove vehicle was driven at a normal driving speed of 30 km/h to 80 km/h depending on the the road type and conditions. All of the experiments were carried out on a PC with a 3.40 GHz i7-4770 CPU. The parameters χ_{XY} and χ_Z in Eqs. (2.19) were set to 0.1 m and 0.3 m respectively, and η in Eqs. (2.20) was set to 10m.

2.5.2 Data Acquisition and Processing

Two different data sets were collected from the Incheon International Airport Express (IIAE) and Seoul National University (SNU) ring road, respectively. The IIAE is

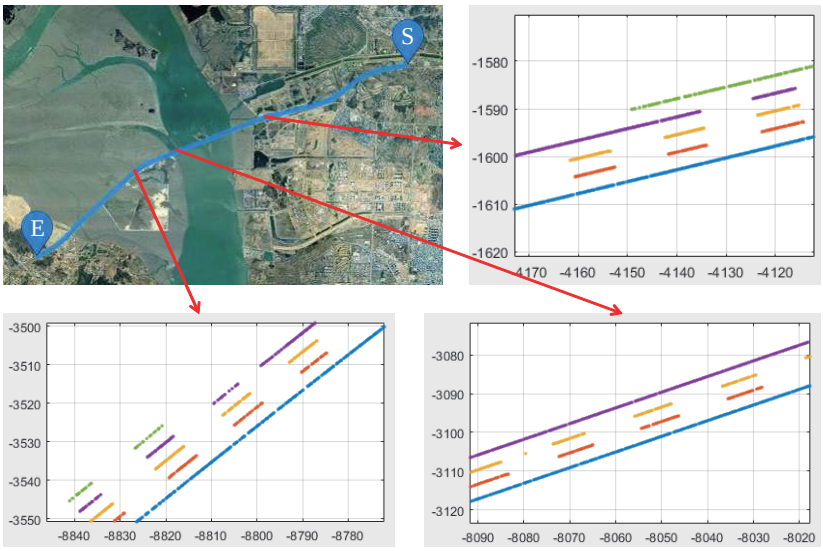
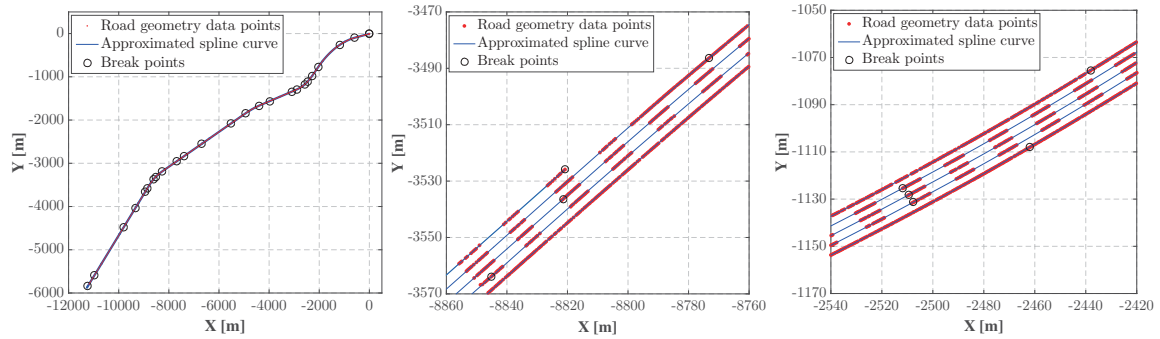


Figure 2.7 Data processing result for the IIAE data set.

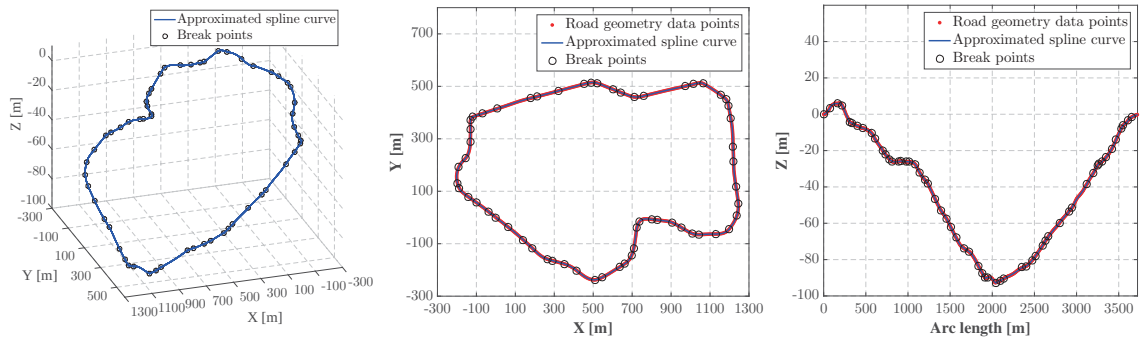
a *type I* road that has four lanes, and thus is suitable to evaluate the road map generation system including the MLS-based data acquisition system. The IIAE data set covers a road length of approximately 13 km and contains about 5,900 frames of synchronized 3D MLS (7.7 billion points) and 6D vehicle pose data. Fig. 2.7 shows a qualitative result of the data processing algorithm when applied to the IIAE data set. From total 7.7 billion laser points, about 65,000 were extracted after executing the data processing algorithm. It is impossible to present the quantitative accuracy of the data acquisition and processing result, since there is no way to measure more accurate coordinates of the lane markings than the RTK-GPS + 3D Lidar configuration that was used in this dissertation. The data processing operates in off-line mode. In the experiment, it took about 38 seconds to process 1 km road of the IIAE data set, where the algorithm was implemented based on the point cloud library [48].

The SNU ring road is a *type II* road and is more appropriate to evaluate the road modeling system since it contains various curves and height variations. The SNU data



(a) (b) (c)

Figure 2.8 The result of the road modeling for the IIAE data set. (a) Overall map, and (b) and (c) enlarged maps.



(a) (b) (c)

Figure 2.9 The result of the road modeling for the SNU data set. (a) 3D road map, (b) road map projected on XY plane, and (c) height of the road.

set consists of a road with a distance of approximately 3.7 km and contains about 6,900 vehicle pose data in intervals of about 0.5 m.

2.5.3 Road Modeling

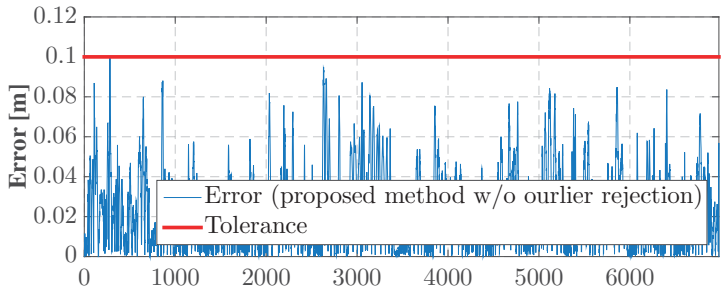
Both the IIAE and SNU data sets were modeled using spline curves by using the proposed road modeling algorithm. Figs. 2.8 and Fig. 2.9 show the results of road

modeling where the XY-plane and Z-axis accuracy thresholds, $D_{\mathrm{max,XY}}$ and $D_{\mathrm{max,Z}}$, were set to 0.1 m and 0.3 m, respectively. Since the IIAE data set contains four lanes, a total of five spline curves were generated, whereas only one spline curve was generated for the SNU data set. The dots in the figures denote the point data, and the lines denote the approximated spline curve. The circles indicate the break points of the spline curves. The road modeling process operates in off-line mode. In the experiment, it took about seven seconds to process 1 km road of the IIAE data set.

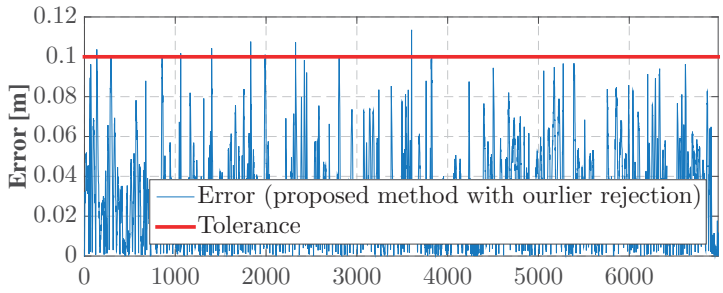
To quantitatively evaluate the performance of the proposed road modeling system in terms of the accuracy, storage efficiency and usability, I conducted a comparison with a B-spline-based state-of-the-art algorithm that was proposed in [14]. The B-spline-based algorithm had been proven to outperform various previous algorithms, including algorithms using polygons, natural cubic splines, or cubic B-splines with a constant interval.

Accuracy

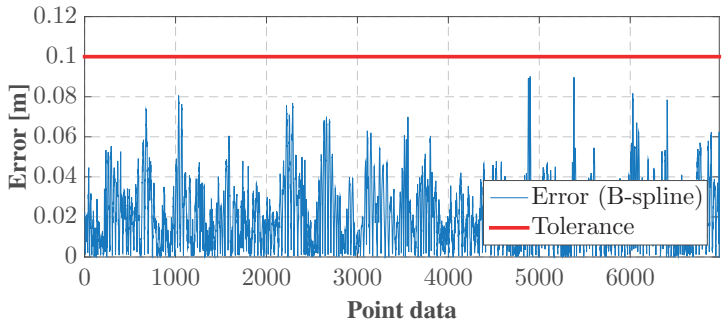
In terms of the accuracy, a major goal of the road modeling system is to generate a curve model for the distance errors between the road geometry data points and the curve not to exceed a preset tolerance, e.g., 0.1 m. Fig. 2.10 shows the modeling accuracy of the proposed algorithm and the B-spline-based algorithm when the SNU data set is used with 0.1m XY-plane accuracy tolerance. The horizontal axis of the figure refers to each data point and the vertical axis refers to the distance error between the data points and the generated curve. Note that two different results are presented for the proposed algorithm depending on whether or not the outlier rejection scheme was used. In the case where the outlier rejection was used, six points were found to violate the 0.1m tolerance. However, exceeding the tolerance does not



(a)



(b)



(c)

Figure 2.10 Errors between road geometry data points and road models generated by (a) proposed algorithm without outlier rejection, (b) proposed algorithm with outlier rejection, and (c) B-spline-based algorithm. Both methods satisfies the 0.1 m accuracy constraint.

necessarily mean that the accuracy is degraded but rather that the outliers were detected and rejected by the algorithm. I found that all points that violated the

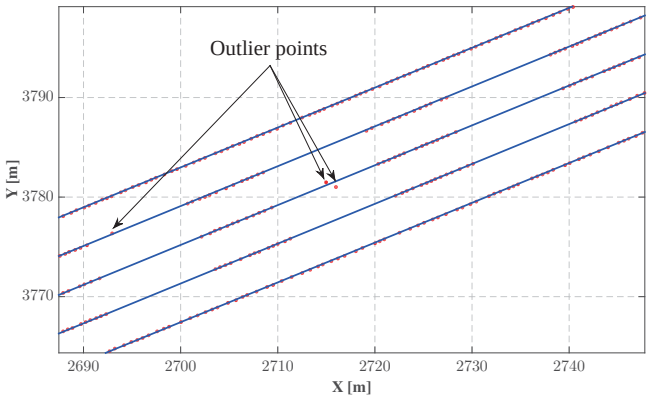
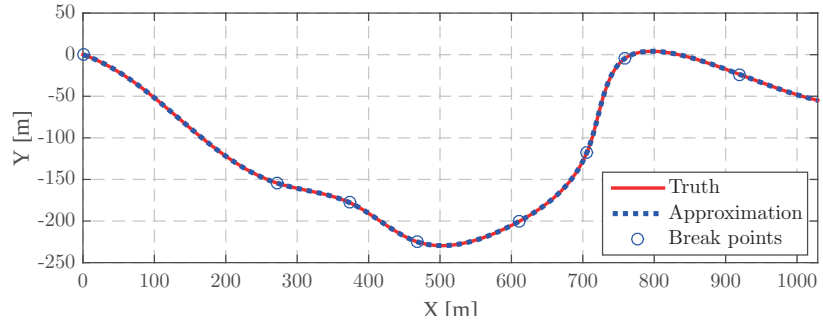


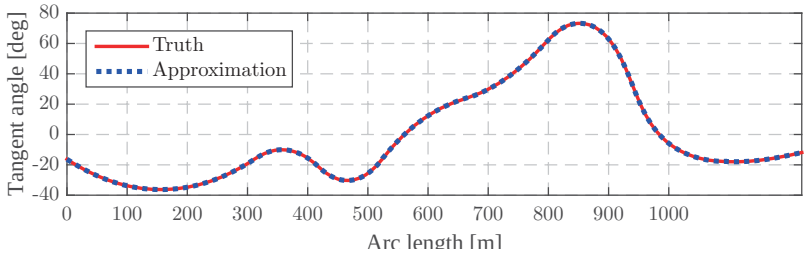
Figure 2.11 Example of the outlier rejection which shows that the outliers were effectively detected and rejected.

tolerance coincide with outliers that had been detected during the road modeling process, as the example in Fig. 2.11 shows. The outlier rejection affects the storage efficiency of the map, and in this example, the entire road was modeled with 55 piecewise polynomials in the case where outliers were rejected while a total of 67 piecewise polynomials were used in the case where outliers were not rejected. As a consequence, we can see that the proposed road modeling algorithm successfully detects and rejects outliers that are contained in data points and also accurately and efficiently represents data points.

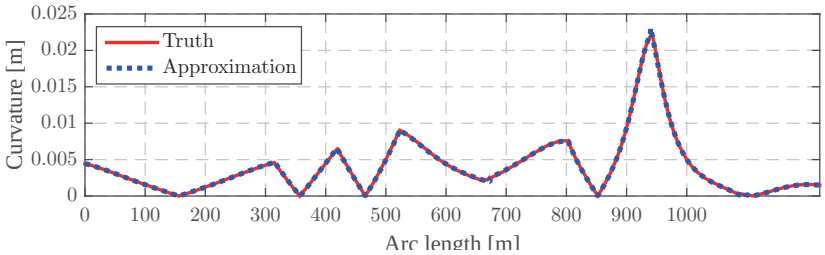
Accurate coordinates as well as accurate tangent angle and curvature information is very important for various intelligent vehicle applications. Such information is essential for safety-related systems, including motion planning systems. Adhering to the tolerance requirements does not necessarily ensure the accuracy of the tangent angle and curvature, and therefore, I also evaluated the road modeling algorithm in terms of the accuracy of the tangent angle and curvature. It is difficult to evaluate the accuracy of the tangent angle and curvature by using real road geometry data points



(a)



(b)



(c)

Figure 2.12 Qualitative result of road modeling in terms of (a) coordinates, (b) tangent angle, and (c) curvature.

since it is impossible to know the exact tangent angle and curvature values of the real road. Thus, I generated artificial road geometry data points for which I know the exact tangent angle and curvature values by first generating an artificial spline curve, shown in Fig. 2.12(a), and then I sampled the curve at every 1 m. To ensure that the data is close to the real values, Gaussian random noise with a standard deviation of

Table 2.4 Accuracy of tangent angle and curvature of the proposed road model and the B-spline-based road model.

		Mean	Std.	RMS	Max
Tangent angle	Proposed	0.898E-02	1.88E-02	1.59E-02	2.67E-01
error [deg.]	B-spline	3.54E-02	3.86E-02	5.24E-02	3.13E-01
Curvature	Proposed	1.67E-05	0.682E-04	0.496E-04	0.841E-03
error [m^{-1}]	B-spline	7.98E-05	1.23E-04	1.47E-04	1.24E-03

0.05m was added to all points. The artificial road geometry data was approximated with a set of polynomials that make use of the proposed road modeling algorithm, and the geometry information that is extracted from the curve was compared to the true value of the road geometry. Fig. 2.12 presents the results, and we can see that the road model accurately expresses all the true road geometries.

A quantitative evaluation was also carried out. Table 2.4 shows the accuracy of the tangent angle and curvature of the road models that were generated using the proposed algorithm and B-spline-based algorithm, where the tolerance of the algorithm was set to 0.1 m. The results show that the proposed algorithm generated a more accurate road model than the B-spline-based algorithm. Furthermore, the levels of maximum error in the tangent angle and curvature were 10^{-1} and 10^{-3} respectively. In general, such levels are acceptable for the errors in intelligent and autonomous vehicle applications.

Storage Efficiency and Usability

The storage efficiency and usability were evaluated together. First, the SNU data set was modeled using the proposed algorithm and the B-spline-based algorithm to com-

Table 2.5 Storage efficiency and usability of the proposed road model and the B-spline-based road model.

	# of segments	# of floating-point numbers	Computational time (μs)			
			Position	Tangent angle	Curvature	Closest point
Proposed	55	715	0.68	0.47	0.59	16
B-spline	178	728	32	82	146	630
B-spline (precomputed derivatives)	178	1805	32	29	52	280

pare the storage efficiency and usability of the two models. In this dissertation, the storage efficiency is measured by the number of floating-point numbers that are used to express the road model. In the case with the proposed set of piecewise polynomial forms, a piecewise polynomial requires thirteen floating-point numbers (one beginning parameter and twelve polynomial coefficients). Therefore, the set of piecewise polynomial forms requires totally $13\times$[*number of piecewise polynomials*] floating-point numbers. The number of required floating-point numbers for the B-spline form depends on the number of knots and control points. Since each control point contains three floating point numbers (X,Y and Z coordinates), the B-spline requires a total of [number of knots]$+3\times$[*number of control points*] floating-point numbers.

The usability of such a system for intelligent and autonomous vehicles is inversely proportional to the computational time required to calculate the essential information from the road models. Four essential pieces of information were selected: position, tangent angle and curvature at an arbitrary point on a curve, and closest point on a curve from an external point. The position, tangent angle and curvature information can be analytically calculated using both the proposed and the B-spline curves. However, the B-spline curve requires complex recursive calculations to obtain this information [40], while the proposed method only requires simple calculations since the first and second derivative of the curve are easily calculated from the polynomials.

The closet point on a curve should be calculated using a numerical method for both curves. In this dissertation, I used Newton's method to evaluate both curves on equal terms (with 10^{-6} tolerance), and the experiments were carried out on a PC with a 3.40 GHz i7-4770 CPU.

Table 2.5 shows the results of the experiment. Note that the B-spline was evaluated using two different modes. In the normal mode, only the knots and control points of the B-spline curve itself were stored in the memory. On the other hand, in the second mode, the knots and control points for the first and second derivatives of the B-spline curve were precomputed and additionally stored in the memory. The precomputation of the derivatives increases the storage space that is required, however, it decreases the computational time. The results show that the proposed algorithm outperforms the B-spline-based algorithm in both modes in terms of both the storage efficiency and usability. In particular, the proposed road model dramatically decreases the computational time needed to calculate the various pieces of information for intelligent and autonomous vehicles. In terms of storage efficiency, note that the B-spline-based algorithm requires a total of 178 curve segments, whereas the proposed algorithm requires only 55 curve segments. This means that, in the case in which the curve is first approximated as a B-spline curve and converted to the piecewise-polynomial form, a large amount of storage is required than when using the proposed algorithm.

2.6 Summary

This chapter presented a road map generation system that simultaneously considers the accuracy, storage efficiency and usability to generate high-precision lane-level road maps for use in intelligent vehicle systems. The overall system is composed of three subsystems, including data acquisition, data processing and road modeling systems.

The MLS data-based data acquisition and processing system efficiently acquires accurate road geometry data by integrating 6D vehicle pose and 3D MLS data. The main contribution of this chapter is to present a road modeling algorithm that models road geometry data as a form consisting of sets of piecewise polynomial curves that are more appropriate for use in intelligent vehicle systems than those of previous road models, such as clothoids or B-splines in terms of usability. The proposed road modeling algorithm maximizes the storage efficiency of the map by minimizing the number of piecewise polynomials needed to express the map. The experiments were conducted to demonstrate that the map that is generated by the proposed algorithm is more effective than those obtained using previous algorithms in terms of the storage efficiency and usability.

Chapter 3

Accurate and Cost-Effective Vehicle Localization in Featureless Environments

Vehicle localization is a key component of the automated driving system. Many automated driving functions such as path planning and motion planning benefit from accurate vehicle position information. Conventionally, integration of GPS and DR has been widely used for vehicle localization [49–52]. However, GPS/DR systems are not appropriate for automated driving, since the GPS/DR systems cannot provide accurate and reliable position information during long time GPS outage.

In order to provide more accurate and reliable position information, localization systems utilizing a map and various perception sensors such as Lidar and camera have been actively studied. In this approach, a precise road map is generated beforehand, and the localization is performed on the map by matching the real-time perception sensor information and the information stored in the map. In terms of the accuracy, the 3D Lidar is one of the most promising sensors for the mapping and localization

problem. The 3D Lidar provides accurate, reliable and plenty 3D range information about surrounding environment, and the performance is not affected by external factors such as intensity of illumination and weather. However, the 3D Lidar is not appropriate to use for commercial vehicles, because of its high cost.

In order for accurate, but cost-effective vehicle localization, in recent years, vision-based vehicle localization has been actively studied [53–58]. The vision-based localization systems use various image features such as lane markings, road markings, and low-level image features for the mapping and localization. More specifically, these feature information are stored in the map, and the feature information extracted from camera images are matched to the stored features to obtain accurate vehicle position. Therefore, the performance of the localization mainly depends on how many features exist on road and in surrounding of the road, and how the features are different for different road sections. Under the environments with rich feature information such as urban environment, there have been many studies on vision-based vehicle localization. However, relatively fewer studies have been conducted for the environment where feature information for the localization is poor, which is named as a featureless environment in this paper. For example, in a highway environment, in many road sections there exist no other feature information except the lane markings.

This paper proposes a precise vision-based/map-aided vehicle localization system for the featureless environment. Data from four different types of sources are integrated for this goal: a low-cost GPS, in-vehicle network sensors, a monocular camera, and a high-precision lane-level road map. Even though this low-cost sensor configuration is desired for automated vehicle systems, it causes a so-called measurement ambiguity problem. We show that a conventional particle filter (PF) cannot solve the ambiguity problem, and propose a particle filter with modified likelihood

function based on a probabilistic particle association to solve the problem. Through experiments using real road data, it is demonstrated that the proposed localization system provides accurate vehicle pose information for automated driving under the featureless environment.

The remainder of this paper is organized as follows. Section 3.1 presents related works. In section 3.2, an overview of the localization system is presented. In section 3.3, overall process of the proposed PF-based localization algorithm is described. In section 3.4, detailed map-image matching process and probabilistic measurement association scheme is presented. The accuracy of the proposed localization system is evaluated in section 3.5 through the experiments, and section 3.6 summarizes this paper.

3.1 Related Works

There have been lots of studies for accurate vehicle localization as shown in Fig. 3.1. Integration of GPS and DR (dead reckoning) has been widely used for vehicle localization [49–52, 59]. The GPS provides global positioning information, however, the update rate is low and the error can be increased significantly by GPS signal outage and multi-path effect for short period of time. The DR estimates the vehicle pose using vehicle motion sensors with high frequency. However, stand-alone DR suffers from long-term error accumulation problem. These complementary characteristics of the GPS and DR make it possible to enhance the localization performance by the integration of the GPS and DR. However, during long time GPS outage, the GPS/DR system cannot correct the diverging DR error. Fig. 3.2 show an example of the visual odometry-based DR. For a highway road, a dense visual odometry algorithm proposed in [60] was applied using monocular camera images with 30 Hz frame rate. The

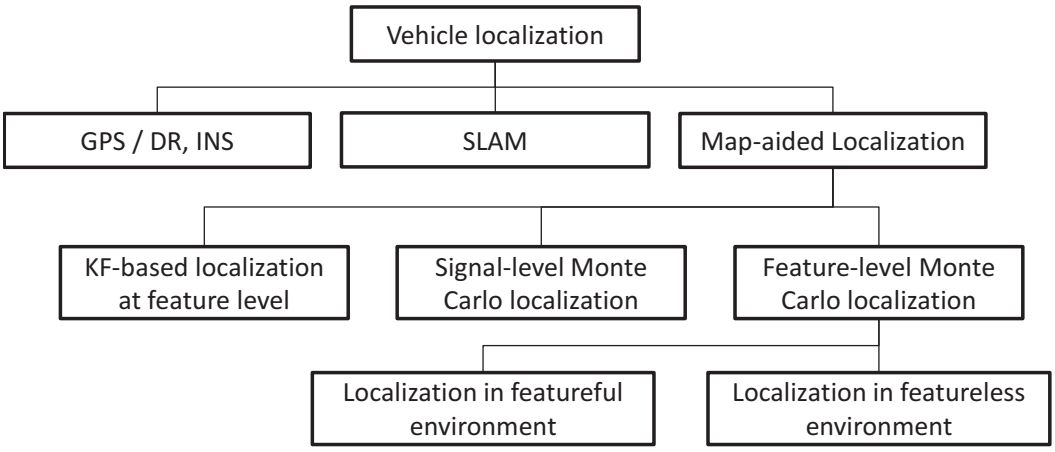
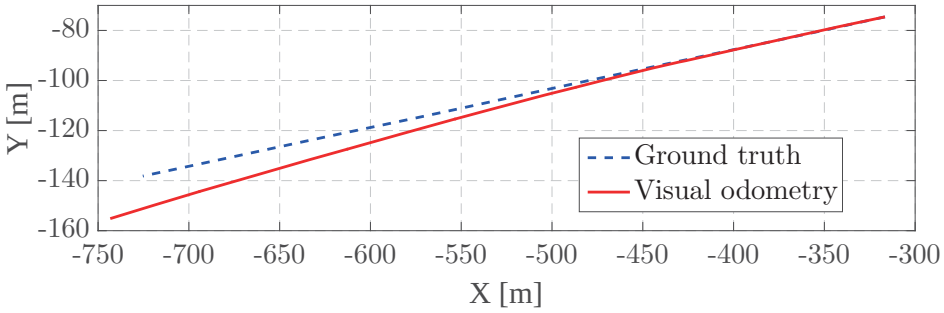


Figure 3.1 Categorization of localization researches.

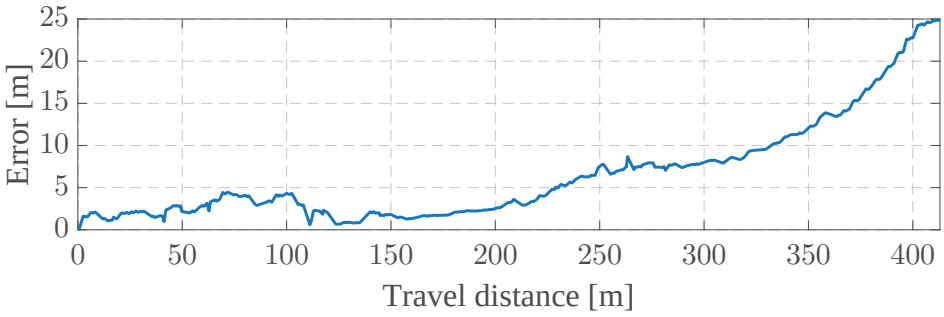
Fig. 3.2a presents the ground truth position of the vehicle and the result of the visual odometry algorithm, and Fig. 3.2b presents the error between the ground truth and the visual odometry result. The result shows that the localization error drastically increases during long time GPS outage.

Simultaneous localization and mapping (SLAM) is a solution for accurate vehicle localization without a GPS. The SLAM generates a map in real-time using various sensors such as laser scanners and camera, and simultaneously localizes the vehicle in the map [61–63]. The SLAM can effectively provide accurate localization result for unknown environment, however, the localization process may get easier if a precise map is given beforehand.

In the map-aided localization approach, a precise and accurate map is given, and the localization is conducted based on the map by utilizing various perception sensors such as laser scanners and cameras. This approach can be categorized into three groups: 1) Kalman filter-based localization at the feature level, 2) signal level Monte Carlo localization, and 3) feature-level Monte-Carlo localization.



(a)



(b)

Figure 3.2 Performance of stand-alone visual odometry.

In the Kalman filter-based localization at the feature level, various features are extracted from various perception sensor measurements, and they are associated with land marks contained in the map. Using the association result, the vehicle position state is updated based on a Kalman filter or its branches [53, 54, 64, 65]. This method has an advantage that the vehicle position can be updated by simple Kalman filter calculations. However, because of the inherent limitation of the Kalman filter, it is not appropriate to apply to nonlinear or non-Gaussian systems.

Because of the limitations of the Kalman filter-based methods, there have been many studies on Monte Carlo localization. In this approach, the nonlinear or non-Gaussian system and measurement models are approximated by a Monte Carlo sampling manner. The Monte carlo localization has two different modes: signal-level local-

ization, and feature-level localization. In the signal-level Monte Carlo localization, the raw measurements of the perception sensors are used for so-called perception-to-map association [66–68]. Since the entire raw measurements are used for data association, complex and inconvenient algorithms for feature extraction are not needed. However, it is inefficient in terms of computational complexity, since the entire raw measurements have to be processed at once.

In order to compensate the disadvantage of the signal-level Monte Carlo localization, recently, feature-level Monte Carlo localization has been actively studied. In this approach, various feature information extracted from the perception sensors are used for perception-to-map association. Various perception sensors including camera, radar, and LIDAR are used to build a precise digital map and for ego-vehicle localization on the map. Among these sensors, camera is most widely used since a camera provides abundant information with low-cost compared to the other sensors. The vision-based localization systems use various image features such as lane markings, road markings, and low-level features for the mapping and localization. Under the environments with rich feature information such as urban environment, there have been many studies on vision-based precise vehicle localization [53–57, 67, 68]. However, relatively fewer studies have been conducted for the environments where feature information for the localization is poor, which named a featureless environment in this chapter. For example, in a highway environment, in many road sections there exist no other feature information except the lane markings. This chapter presents a vehicle localization system for the featureless environments.

3.2 System Overview

3.2.1 Test Vehicle and Sensor Configuration

A test vehicle was configured based on various low-cost sensors for the vehicle localization. First, I utilized some vehicle motion sensor information that the electronic stability control (ESC) module of the vehicle contains. These sensor information can be easily captured from the in-vehicle network (controller area network, CAN). Two types of sensor information (wheel speed sensors and a yaw rate sensor) were used for dead reckoning of the vehicle, and the specifications of the motion sensors are listed in Table 3.1.

A low-cost GPS was used for basic positioning information. Specifically, the GPS positioning information was used to initialize the localization system, and to prevent the localization error from diverging. I installed JYSYSTEM B20, and the specifications of the GPS are given in Table 3.2. The B20 support the DGPS mode, however, the DGPS mode was turned off in this configuration.

The test vehicle equipped with a monocular camera. The camera provides images of a 90 degree field of view (FOV) in the front of the vehicle. Multiple cameras can be used, however, I found that single camera provides enough localization accuracy and it is favorable to minimize the computational burden. The detailed specifications of the camera are listed in Table 3.3.

3.2.2 Augmented Road Map Data

The road map presented in the previous chapter represents the road lines as a set of piecewise polynomials. In other words, the road map only contains basic road geometry information such as position coordinates, tangent angle, and curvature. However, this basic road geometry information is insufficient to utilize for precise

Table 3.1 Specification of in-vehicle network sensors.

Sensor	Range	Frequency	Resolution	Noise RMS
Wheel speed sensor	0∼130	50 Hz	0.035 m/s	0.3 m/s
Yaw rate sensor	120	50 Hz	0.0625 deg/s	0.5 deg/s

Table 3.2 Specification of low-cost GPS.

Position accuracy	2.5m circular error probability (CEP)
Update rate	1∼10 Hz

Table 3.3 Specification of monocular camera.

Field of view	Frame rate	Resolution
90°	30 Hz	1280×1024

vehicle localization. Therefore, the road map is augmented so that it contains more information that is useful for precise vehicle localization. Two types of information are additionally included. First one is dashed lane marking information. A large portion of real roads contains dashed lane markings. Therefore, the dashed lane marking information can increase both the lateral and longitudinal vehicle localization performance in many road environments. Second is road markings information. Various road markings such as arrows and texts on the roads can be used as landmarks for the localization. Especially, the road marking information is useful for increasing the longitudinal localization performance in solid line road sections.

In order to include the dashed lane marking information in the road map, I first found the points corresponding to the beginning and end point of the lane marking

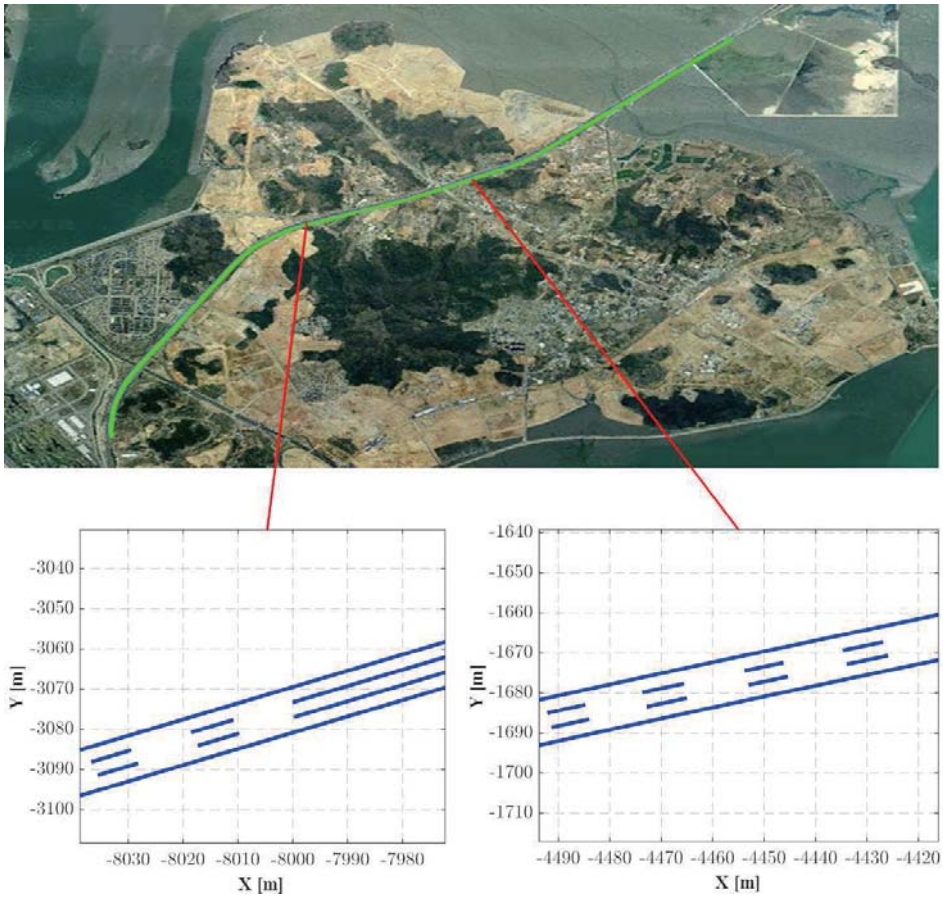


Figure 3.3 Road map including dashed lane marking information.

using the road geometry point data obtained in chapter 2. Then a set of pair of points denoting the beginning and end points of the lane markings is obtained for a road line, i.e.,

$$LMp^i = \left\{ \{(x_b, y_b), (x_e, y_e)\}_j^i, j = 1, ..., N_i \right\}, i = 1, ..., N_{\text{lane}}, \tag{3.1}$$

where i denote the index of the road lines and j means the index of the lane markings. Next, in order to apply this information to the polynomial-based road map, the curve parameters (arc-length parameters) are founded and the sets of parameters

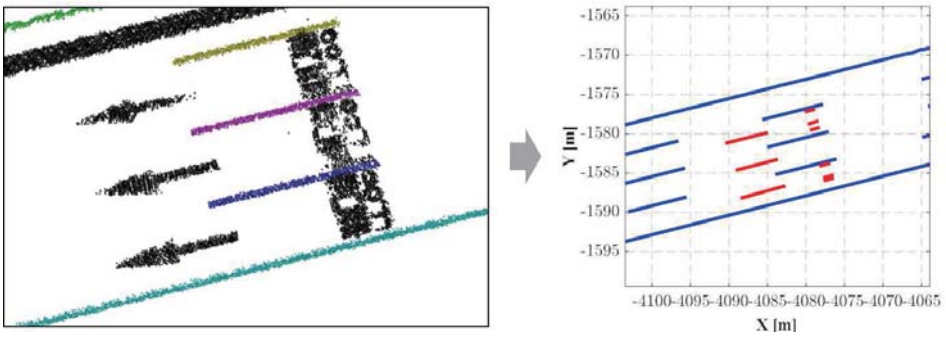


Figure 3.4 Road map including road marking information.

representing the beginning and end point of the lane markings are generated as below:

$$LMpam^i \left\{ \{s_b, s_e\}_j^i , j = 1, ..., N_i \right\}, i = 1, ..., N_{\text{lane}}, \tag{3.2}$$

where $\{s_b, s_e\}_j^i$ denote the curve parameters corresponding to the beginning and end point of the lane markings. From this approach, the dashed lane marking information can be easily extracted from the road map as shown in Fig. 3.3.

The road marking information is extracted manually from the LIDAR data as shown in Fig. 3.4. As will be explained later, the road map data should be transformed to an image plane for map to image matching. Therefore, to reduce the computational burden required for the transformation process, only the vertical line components of the road markings are extracted from the LIDAR data. Extracted road markings are stored in the road map by assigning the index number of the corresponding curve segment to each road marking. Finally, the road marking information can be easily extracted from the map as shown in Fig. 3.4.

Including the additional information in the map may increase the data size. However, the dashed lane markings are represented as compact forms as explained above. For the Incheon international airport expressway, I examined the required data to build the map. For the original map that does not contain the dashed lane marking

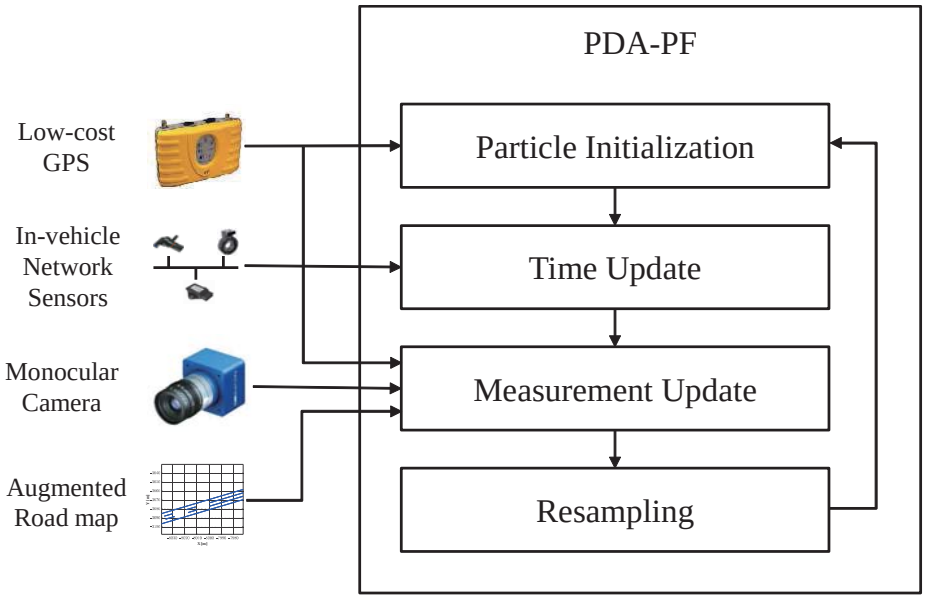


Figure 3.5 Overall localization system architecture.

information, average 700 bytes data per 1 km road was required for a lane. For the augmented map, average 1.4 Kbytes data per 1 km road was additionally required for a lane.

3.2.3 Vehicle Localization System Architecture

The overall architecture of the proposed localization system is described in Fig. 3.5. There are four types of data sources for the vehicle localization: low-cost GPS, in-vehicle network sensors, monocular camera, and augmented road map. The four types of data are integrated by a particle filter to estimate accurate 3D vehicle pose (position, heading). The low-cost GPS is used to initialize particles of the filter and to prevent the localization error from diverging at the measurement update step. The in-vehicle network sensor information (wheel speed, yaw rate) is used to capture the vehicle motion within a short period of time. Specifically, the motion sensors are used

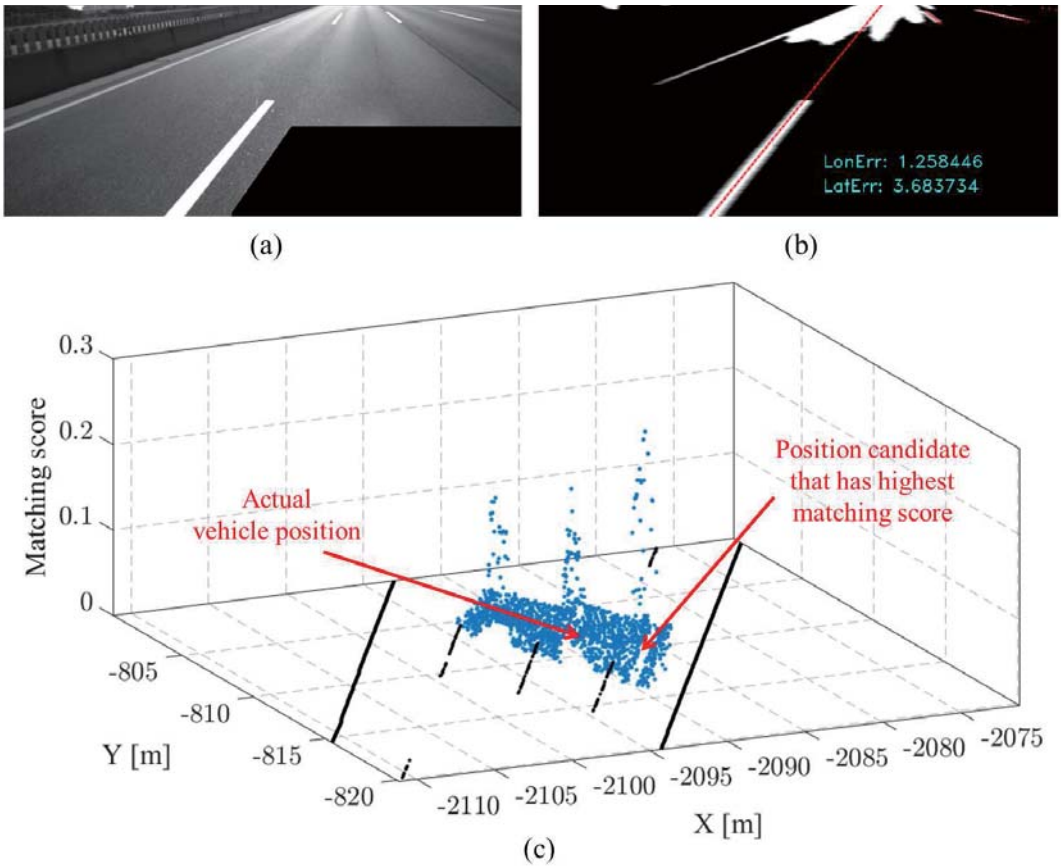


Figure 3.6 Example of the matching ambiguity problem.

to predict the motion of particles at the time-update step. The road map and image are the key information for accurate localization. The lane marking and road marking information contained in the map are matched to the information that the image contains, and the particles are updated using the matching results. In the rest of this chapter, detailed localization algorithm is described.

3.2.4 Problem Statement

The core of the proposed localization algorithm is on estimating accurate pose of the vehicle only using simple road marking features (lane markings and other road mark-

ings appearing intermittently) stored in the map. Lack of these feature information may cause an ambiguity problem. At each measurement update step of the particle filter, a camera image and a map data around the vehicle are given. The problem is to find an optimal vehicle pose that maximize the matching score between the image and the map data. If there is abundant feature information in the map, there exist only one pose solution that has significantly high matching score compared to the other pose candidates. However, in our problem, there exist multiple candidates of solution that have similarly high matching scores due to the ambiguity of the road marking feature information. Fig. 3.6 shows an example of this problem. Fig. 3.6(c) shows the matching score with respect to the vehicle pose candidates distributed around the actual vehicle position, and Fig. 3.6(a) and Fig. 3.6(b) show the map-image matching result for the pose candidate that has highest matching score. As we can see from this example, the ambiguity in the map-image matching can cause large errors that cannot be acceptable for automated driving.

The conventional particle filter cannot handle this ambiguity problem, since it doesn't care of the possibility that the measurement was distorted by the ambiguous problem. In this context, this chapter proposes a particle filter with modified likelihood function to handle the ambiguity problem.

3.3 Particle filter-based Vehicle Localization Algorithm

Fig. 3.7 shows the primary process of the localization algorithm based on the particle filter. At the initialization step, the state and weight values of the particles are initialized based on the GPS positioning information. After finishing the initialization, the recursive filtering process that is composed of the time update and measurement update is performed. At the time update step, the probabilistic distribution of the

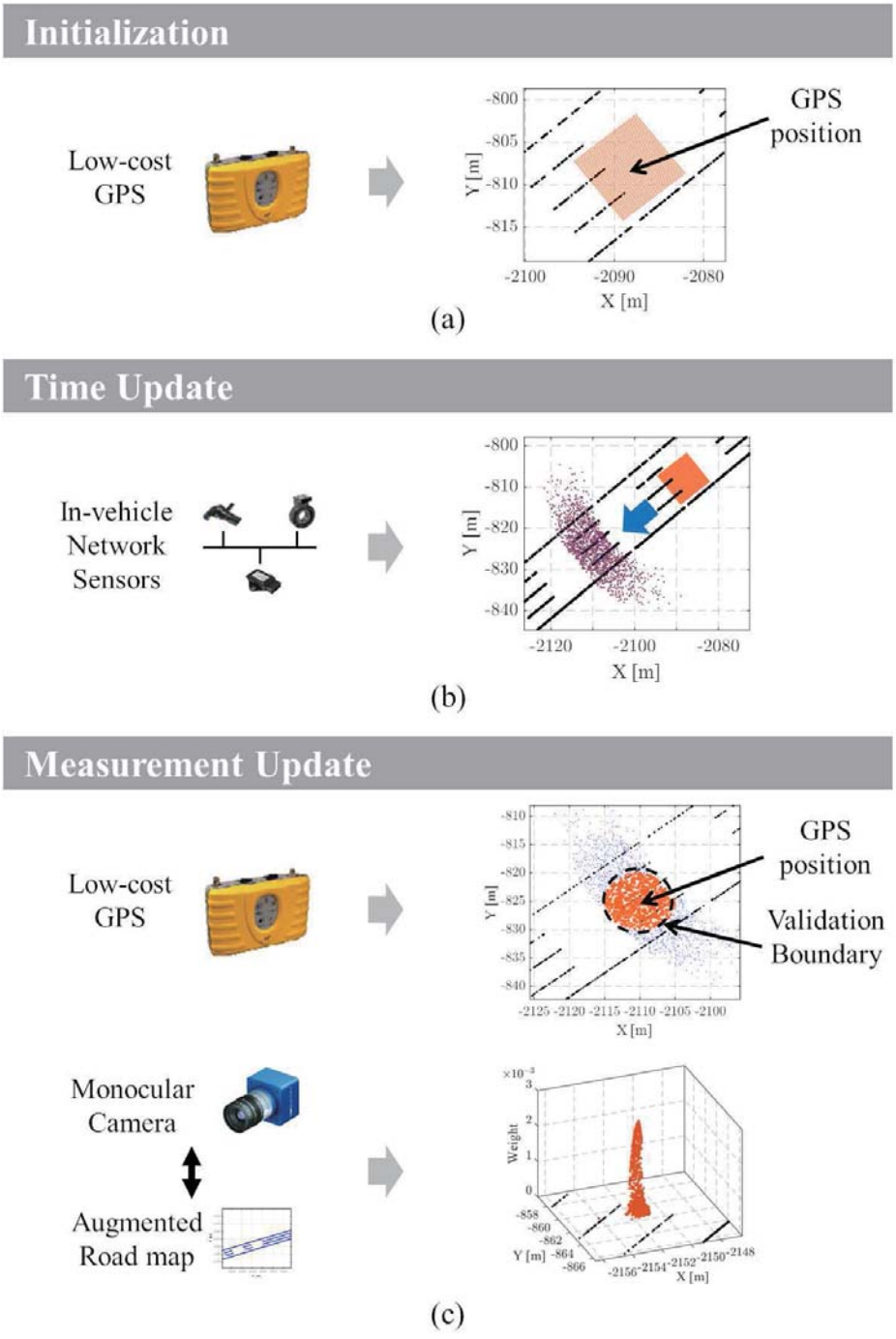


Figure 3.7 Primary particle filter process for vehicle localization.

vehicle pose is predicted by updating the state values of the particles according to the vehicle motion model. At the measurement step, the GPS and map-image measurements are used to update the posterior distribution of the vehicle pose.

3.3.1 Initialization

The initialization is performend based on the GPS positioning information. If an initialization command is issued, the initialization module wait until the healthy GPS positioning information is received. Healthy status of the GPS information is checked according to various GPS signal condition parameters such as number of satellites and horizontal dilution of precision (HDOP). If the status of the GPS information is healthy, initial particles are generated around the GPS position. I assume that the initial vehicle pose $p(x_{0|0})$ is uniformly distributed. According to this assumption, the particles are uniformly generated with rectangular-shape around the GPS position as shown in Fig. 3.7(a), and the weights of the particles are uniformly normalized as follows:

$$w^i = \frac{1}{N}. \tag{3.3}$$

The size of the rectangle depends on the error characteristics of the GPS receiver. I found that 9 m by 9 m size of rectangle have worked well in our experiment.

3.3.2 Time Update

In the time update step, the state of each particle is updated based on the simple vehicle motion model as

$$\begin{aligned} x^i_{k|k-1} &= \begin{bmatrix} \psi^i_{k|k-1} \\ X^i_{k|k-1} \\ Y^i_{k|k-1} \end{bmatrix} \\ &= \begin{bmatrix} \psi^i_{k-1|k-1} + \Delta T \cdot r^i_{k-1} \\ X^i_{k|k-1} + \Delta T \cdot V^i_{k-1} \cos\left(\psi^i_{k-1|k-1} + \frac{1}{2}\Delta T \cdot r^i_{k-1}\right) \\ Y^i_{k|k-1} + \Delta T \cdot V^i_{k-1} \sin\left(\psi^i_{k-1|k-1} + \frac{1}{2}\Delta T \cdot r^i_{k-1}\right) \end{bmatrix}, \end{aligned} \tag{3.4}$$

where r^i and V^i are control inputs corresponding to yaw rate and wheel speed, respectively. The r^i and V^i are random variables rather than deterministic values derived from below error models:

$$\begin{aligned} r^i_k &= r_{\mathrm{GYRO},k} + \varepsilon^i_{\mathrm{GYRO}}, \\ V^i_k &= V_{\mathrm{WHEEL},k} + \varepsilon^i_{\mathrm{WHEEL}}, \end{aligned} \tag{3.5}$$

where $r_{\mathrm{GYRO},k}$ and $V_{\mathrm{WHEEL},k}$ are yaw rate and wheel speed sensor values obtained from the in-vehicle network, and $\varepsilon^i_{\mathrm{GYRO}}$ and $\varepsilon^i_{\mathrm{WHEEL}}$ are corresponding Gaussian errors.

3.3.3 Measurement Update

The weight of each particle w^i_k is updated by evaluating the likelihood of measurements $p(z_k|x^i_{k|k-1}, m)$ conditioned on the time-updated particle state and the map as follows:

$$w^i_k \propto w^i_{k-1} p(z_k|x^i_{k|k-1}, m). \tag{3.6}$$

There are two types of measurements to update the state: GPS position information and a map-image matching measurement. The GPS measurement is responsible for preventing the localization error from diverging. Accurate state correction is performed based on the map-image measurement. This section describes details of the two measurement models and update procedures.

GPS Measurement Update

Generally, the error of the low-cost GPS position data can be modeled as a combination of a Gaussian random noise and a bias error as below:

$$\varepsilon_{GPS} = \varepsilon_{\sigma} + \varepsilon_{b}. \tag{3.7}$$

Even though the magnitude of the bias error varies slowly, it is difficult to model and estimate the magnitude and direction of the error. This nature makes it difficult to use the GPS position information as a measurement of a Bayesian filter. However, fortunately, since the maximum magnitude of the bias error is regulated below few meters under healthy GPS signal environment, we can roughly model the GPS error as a circle-shaped uniform distribution with a radius δ_{GPS}. Then the likelihood function with respect to the GPS measurement $y_{\mathrm{GPS,k}}$ and a particle $x^{i}_{k|k-1}$ is defined as follows:

$$q^{i}_{\mathrm{GPS}} = \begin{cases} 1, & \text{if } \left(\mathrm{X^{i}_{k}} - \mathrm{X_{GPS,k}}\right)^2 + \left(\mathrm{Y^{i}_{k}} - \mathrm{Y_{GPS,k}}\right)^2 \leq \delta^2_{\mathrm{GPS}} \\ 0, & \text{otherwise} \end{cases}. \tag{3.8}$$

Map-Image Measurement Update

The likelihood of the map-image measurement q^{i}_{CAM} is obtained by matching the road map data to the camera image based on each particle state as below:

$$q^{i}_{\mathrm{CAM},k} = p(z_k | x^{i}_{k|k-1}, m). \tag{3.9}$$

The likelihood function for the map-image measurement is the most important part of this chapter. Therefore, I will present more detailed algorithms for the measurement update and likelihood function derivation in Section 3.4.

3.3.4 Integration

Since GPS and map-image measurements are independent, using Eqs. 3.6, the weight is updated as follows:

$$w_k^i = w_{k-1}^i \cdot q_{\mathrm{GPS},k}^i \cdot q_{\mathrm{CAM},k}^i. \tag{3.10}$$

After updating the weights, a normalization process is performed as follows:

$$w_k^i = \frac{w_k^i}{\sum_{j=1}^{N} w_k^i}. \tag{3.11}$$

3.3.5 State Estimation

After updating the weights, we can approximate the posterior distribution with respect to the vehicle position as follows:

$$p(x_{k|k}|z_{1:k}, m) \approx \sum_{i=1}^{N} w_k^i \delta(x - x_{k|k-1}^i), \tag{3.12}$$

where $\delta(x - x_k^i)$ is a delta function centered at x_k^i. The most common way to estimate the state value from the posterior distribution is applying the minimum mean square error (MMSE) estimate, which can be approximated by the weighted mean of the particle density distribution, as follows:

$$\hat{x}_{k|k} = \sum_{i=1}^{N} w_k^i x_{k|k-1}^i. \tag{3.13}$$

However, in many cases, the likelihood distribution of the map-image measurement has a multimodal distribution as shown in Fig. 3.6. In a multimodal distribution, it

is better to use a maximum a-posteriori estimate as follows:

$$\hat{x}_{k|k} = \arg\max_{x^i_{k|k-1}} p\left(x^i_{k|k-1}|z_{1:k}, m\right). \tag{3.14}$$

3.3.6 Resampling

The resampling process is conducted to prevent the particles from being concentrated in specific areas. The degree of the particle concentration can be measured by

$$\hat{N}_{\text{eff}} = \frac{1}{\sum_{i=1}^{N} (w^i_k)^2}. \tag{3.15}$$

$\hat{N}_{\text{eff}}$ approximates the effective number of samples. In this dissertation, if $\hat{N}_{\text{eff}} < 2/3N$, the resampling is conducted based on a systematic resampling method. This method is efficient in terms of computational time.

3.4 Map-Image Measurement Update with Probabilistic Data Association

This section presents the map-image measurement update process. The goal of this step is to get a likelihood distribution of the image measurement conditioned on the particle states and the map. In common with the previous Monte-carlo localization algorithms, proposed algorithm calculates the likelihood value with respect to each particle by calculating the matching score between given camera image and the road map data conditioned on the particle state. However, in order to handle the ambiguity problem of the measurement, the probabilistic data association scheme is adopted, and the likelihood values are updated by so-called association probability.

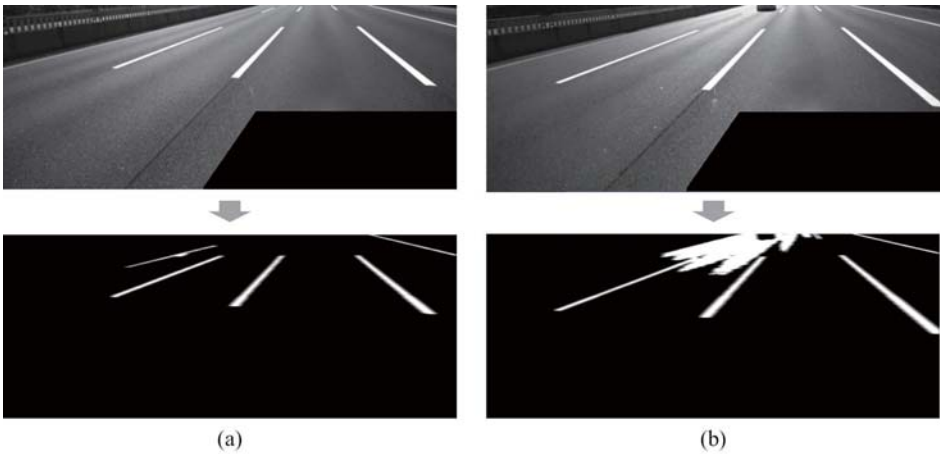


Figure 3.8 Example of the intensity thresholding of images.

3.4.1 Lane Marking Extraction and Measurement Error Model

Road Marking Extraction

In order to match the feature from the camera image to the map data, road marking components are extracted from the camera image. There exist a lot of algorithms for road marking extraction, however, simple intensity thresholding method is used in this dissertation as shown in Fig. 3.8. The reason why simple intensity thresholding method is used is as follows: 1) By simplifying the image processing, we can reduce overall computational time. 2) The identification of the road markings is not needed for the matching algorithm. 3) Some false positives and false negatives are acceptable for the matching algorithm.

Measurement Error Modeling

In this dissertation, only three vehicle pose values (X, Y, and heading) are estimated by assuming that the height, pitch, and roll are constant. This is because the required number of the particles should be exponentially increased by the dimension of the

state vector. In order to reduce the dimension of the state vector, first it is assumed that the road is on a flat plane. This means that the height is constant. However, it is difficult to set the pitch and roll of the vehicle as a constant since they vary frequently in dynamic driving conditions. In order to handle the variation of the pitch and roll angles, the pitch and roll errors are reflected in the measurement (image) by applying the Gaussian filter to the image as shown in Fig. 3.9.

Map to Camera Matching

After finishing the road marking extraction and applying the Gaussian noise, map to camera matching is performed. First, the road map data is transformed to the image plane with respect to each particle state (pose candidate). The transform contains some matrix multiplications and non-linear arithmetic operations. Therefore, in order to reduce the computational burden for real-time operation, the end-points of the road markings are transformed to the image plane, and then the end-points are interpolated in the image plane as shown in Fig. 3.10.

After finishing the coordinate transformation, the measurement likelihood

$$q^i_{\mathrm{CAM},k} = p(z_k|x^i_{k|k-1}, m) \tag{3.16}$$

is calculated by calculating the matching score as follows:

$$p(z_k|x^i_{k|k-1}, m) \propto \exp\left[\frac{1}{N_{p,j}}\sum_{j=1}^{N_{p,j}} -\left(\frac{(I^i_{\mathrm{Cam},j,k} - I^i_{\mathrm{Map},j,k})^2}{2\sigma_I^2}\right)\right], \tag{3.17}$$

where $I_{\mathrm{Map,j}}$ denotes the intensity value of jth pixel corresponding to the map data, $I_{\mathrm{Cam,j}}$ denotes the intensity of image pixel corresponding to the image pixel coordinates of the I_j, and the $N_{p,j}$ means the number of pixels corresponding to the map data. Here, $I_{\mathrm{Map,j}}$ is set to 255 which is the maximum intensity value. Eqs. 3.17 can

(a) Stdv = 0

(b) Stdv = 11

(c) Stdv = 21

Figure 3.9 Example of the Gaussian filtering.

be derived by assuming that the likelihoods of pixels are independent and follow the Gaussian distribution as below:

$$p(z_k|x^i_{k|k-1}, m) = \prod_{j=1}^{N_{p,j}} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-(I^i_{Cam,j,k} - I^i_{Map,j,k})^2}{2\sigma_I^2}\right). \tag{3.18}$$

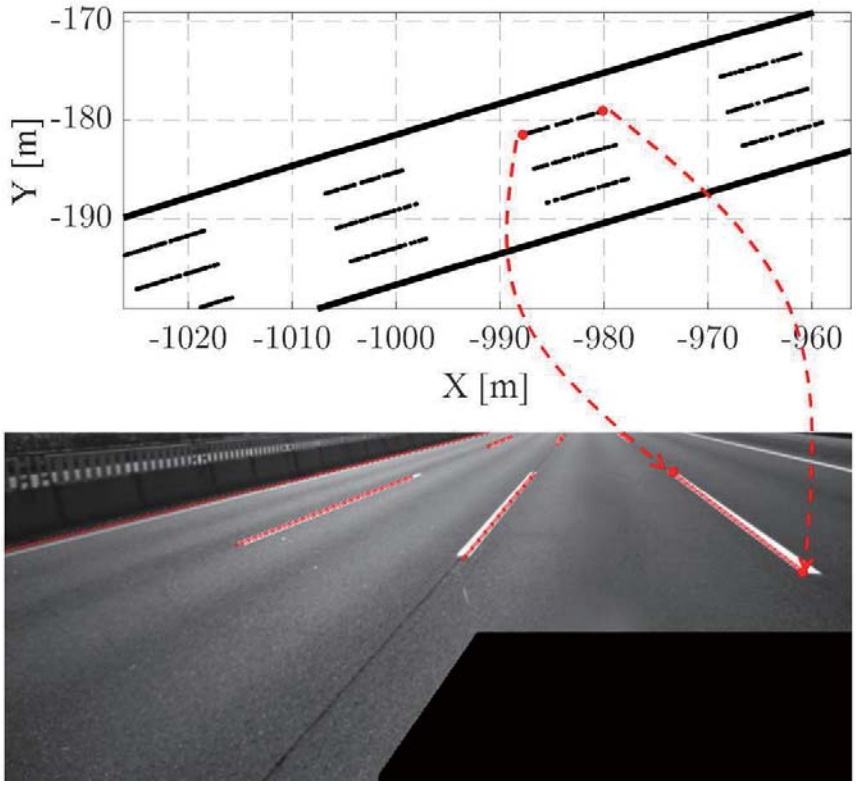


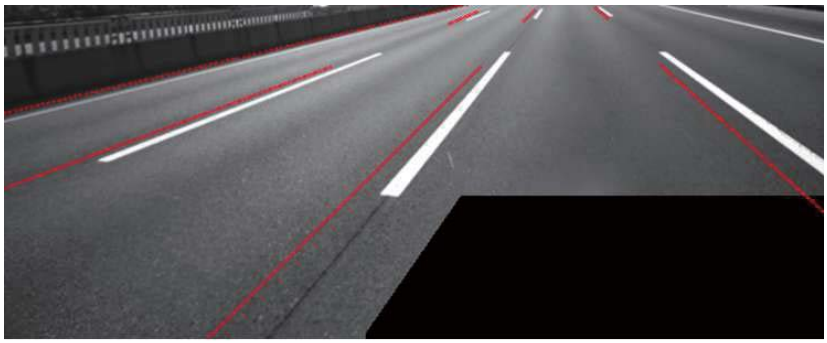
Figure 3.10 Example of the map to image plane transformation.

The likelihood function 3.17 is intuitive in terms of the fact that the likelihood value is equal to 1 if the map data and the image features are perfectly matched, and close to 0 if the map data and the image features are not matched at all.

Fig. 3.11 shows the example of the map-image matching results and corresponding likelihood values for three different particles.

Measurement Association

The measurement model and the likelihood function explained in the previous sections is vulnerable to the measurement ambiguity problem. This section presents the measurement association scheme to handle the ambiguity problem.



(a) Likelihood value = 0.0247

(b) Likelihood value = 0.341

(c) Likelihood value = 0.160

Figure 3.11 Example of the Gaussian filtering.

We can think that the likelihood function obtained in the previous section is based on the hypothesis that all the measurements certainly came from the true state.

Consequently, the likelihood function ignores the hypothesis that the measurement came from false state due to the ambiguity problem. In order to tackle this problem, two mutually exclusive hypotheses are considered as follows:

- θ_0^i : The hypothesis that the measurement is not associated with ith particle.
- θ_1^i : The hypothesis that the measurement is associated with ith particle.

Using these events and the total probability theorem, we can write $p(z_k|x_{k|k-1}^i, m)$ as

$$p(z_k|x_{k|k-1}^i, m) = \sum_{j=0}^{1} p(z_k|x_{k|k-1}^i, m, \theta_{j,k}^i) \cdot p(\theta_{j,k}^i|x_{k|k-1}^i, m). \tag{3.19}$$

Conditioned on the association event we can evaluate $p(z_k|x_{k|k-1}^i, m, \theta_{j,k}^i)$ as

$$p(z_k|x_{k|k-1}^i, m, \theta_{j,k}^i) = \begin{cases} 0 & , j = 0, \\ \exp\left[\frac{1}{N_{p,j}} \sum_{j=1}^{N_{p,j}} -\left(\frac{(I_{\text{Cam},j,k}^i - I_{\text{Map},j,k}^i)^2}{2\sigma_I^2}\right)\right] & , j = 1. \end{cases} \tag{3.20}$$

Here, for simplicity, it is assumed that, for the false detection event, the likelihood is zero.

$\beta_{j,k}^i = p(\theta_{j,k}|x_{k|k-1}^i, m)$ is named as the association probability in this dissertation. Interesting point is that the association probability is independent to the measurement z_k. It is only conditioned on the particle state $x_{k|k-1}^i$ and the map data m. Additionally, I assume that the measurement association event is independent to the map data, and then the association probability can be written as

$$\beta_{j,k}^i = p(\theta_{j,k}^i|x_{k|k-1}^i). \tag{3.21}$$

Since I assumed that the likelihood is zero for the false detection event, the association probability for true detection, $\beta_{1,k}$, is only our interest. $\beta_{1,k}$ can be expressed as

$$p(\theta_{1,k}|x_{k|k-1}^i) = \frac{p(x_{k|k-1}^i|\theta_{1,k}^i)p(\theta_{1,k}^i)}{p(x_{k|k-1}^i)} \tag{3.22}$$

by the Bayes theorem. Here, the prior probability $p(\theta^i_{1,k})$ is constant over the particles, and $p(x^i_{k|k-1})$ are i.i.d, we can conclude that

$$p(\theta_{1,k}|x^i_{k|k-1}) \propto p(x^i_{k|k-1}|\theta_{1,k}) \sim \mathcal{N}(x^i_{k|k-1}; x_{k|k-1}, Q_k). \tag{3.23}$$

Finally, the likelihood of map-image measurement is calculated by

$$p(z_k|x^i_{k|k-1}, m) = \exp\left[\frac{1}{N_{p,j}}\sum_{j=1}^{N_{p,j}}\left(\frac{(I^i_{\text{Cam},j,k} - I^i_{\text{Map},j,k})^2}{2\sigma_I^2}\right)\right] \cdot \mathcal{N}(x^i_{k|k-1}; x_{k|k-1}, Q_k). \tag{3.24}$$

3.5 Experimental Validation

3.5.1 Experimental Environments

Test Road

The proposed localization system was evaluated at the Incheon international airport express (IIAE) in Korea. From the airport tollgate to Geumsan interchange, about 13 km section was selected for evaluation as shown in Fig. 3.3. The IIAE does not contain enough features for localization except the lane marking information. Therefore, it is appropriate to evaluate the proposed localization system. In addition, IIAE includes an about 3 km of solid line road section, which is useful for evaluating the longitudinal localization performance.

Algorithm Implementation

The proposed particle filter-based localization was implemented using 1600 particles. The execution time is about 20∼30 ms based on Intel i7 CPU computing unit. I found that more than 1600 particles do not improve the localization performance much.

Monte-Carlo Experiments

Since the proposed localization algorithm is based on the random sampling, the result may different for each trial. Therefore, it is not reliable to evaluate the performance of localization by using the results of only one of the experiments. To reliably evaluate the algorithm based on random sampling, the evaluation uses the Monte-Carlo method, which repeats the experiment several times with different random samples and the same control inputs. In this dissertation, 10 trials for each experiment were executed, and the results are averaged to get final results.

3.5.2 Localization Accuracy

Comparison to GPS+DR localization

In order to evaluate the performance of the proposed localization, first, a comparison study performed with the GPS+DR-based localization. The GPS+DR-based localization algorithm integrates the low-cost GPS position information and the in-vehicle network sensors using an extended Kalman filter. A RTK-GPS with centimeter-level accuracy was used to obtain a ground-truth vehicle position data, and the localization errors were evaluated based on the ground truth data. Position error, longitudinal position error, and lateral position errors of two algorithms were evaluated, and these errors were obtained by averaging 10 trials of Monte-Carlo experiments. Fig. 3.12 shows the comparison result. We can see that the longitudinal errors are maintained under 1 m in normal conditions, however, in the section between $t = 200$ and $t = 300$, the error get increased remarkably. This section is solid lane marking section in which the longitudinal features do not exist. However, the maximum error is limited below 4 m. This is because the GPS measurement update prevents the error from diverging. We can see the excellence of the proposed algorithm in terms of the lateral local-

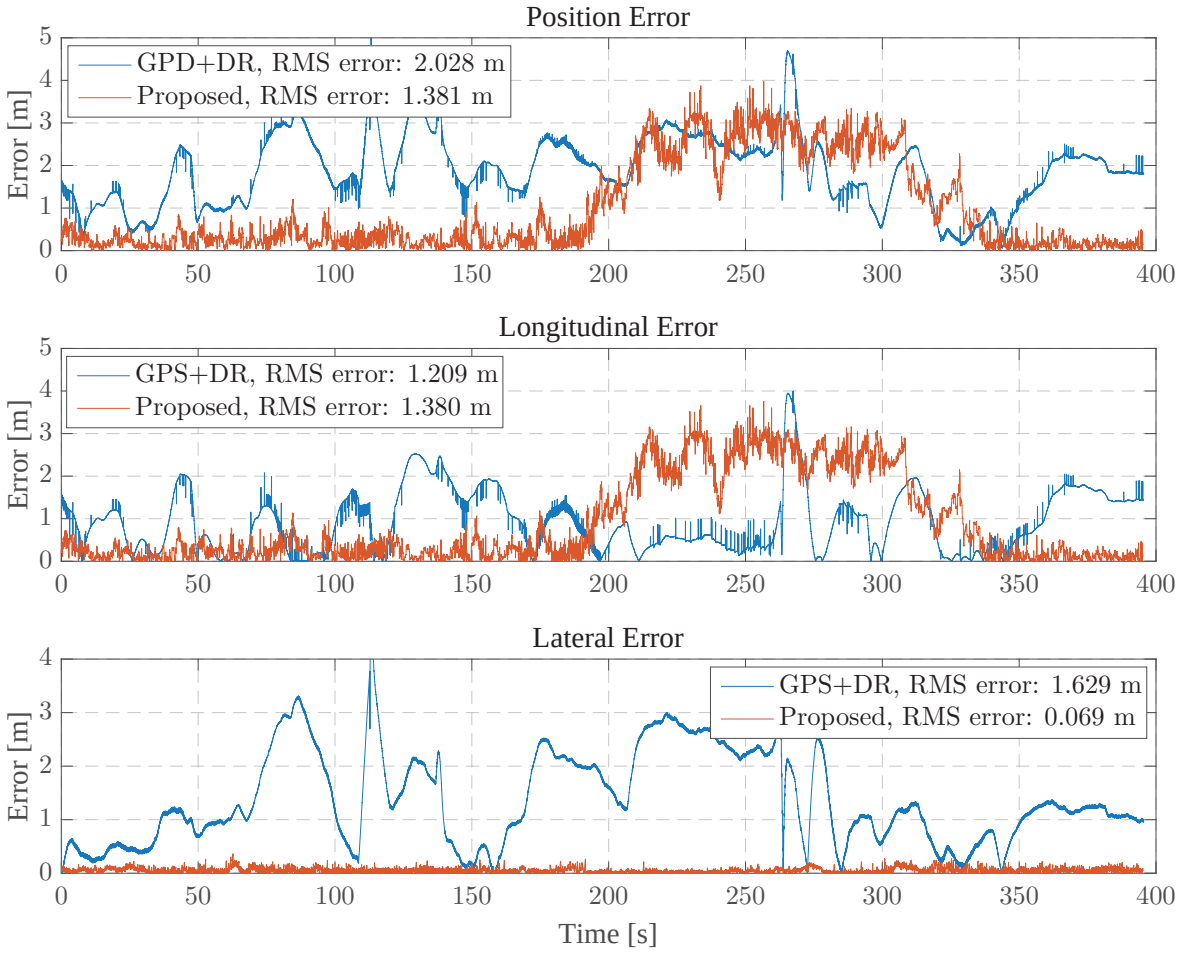


Figure 3.12 Error comparison result of a GPS+DR-based localization algorithm and the proposed localization algorithm.

ization accuracy. The proposed algorithm provides very accurate lateral localization accuracy with a RMS error of about 7 cm. Therefore, from the experiment result, we can see that the proposed algorithm can provide accurate and reliable positioning information which is sufficient for autonomous vehicle driving.

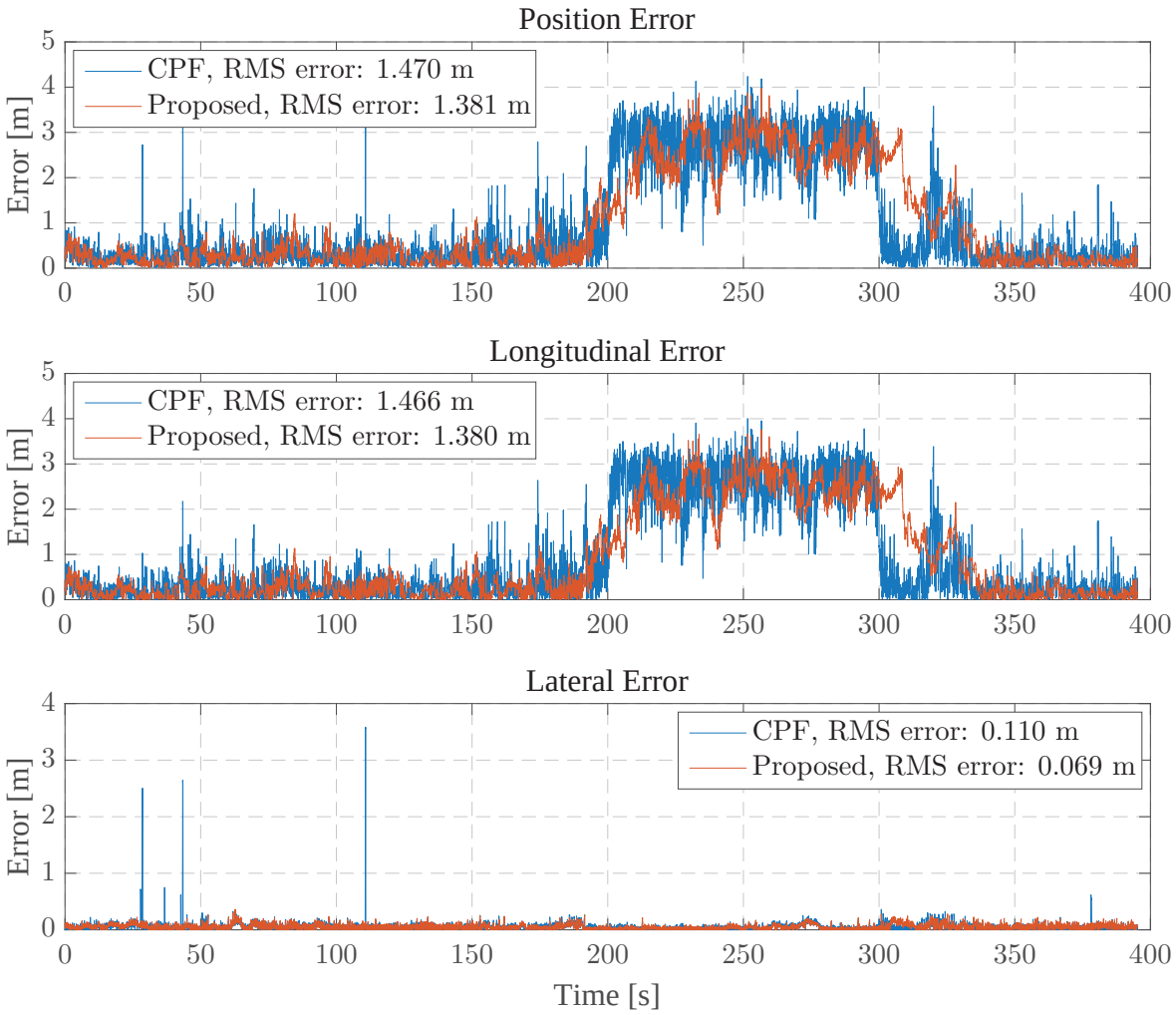


Figure 3.13 Error comparison result of a CPF-based localization algorithm and the proposed localization algorithm.

3.5.3 Effect of the Probabilistic Measurement Association

In order to evaluate the performance of the proposed localization algorithm, a comparison study was performed with conventional particle filter (CPF) with a conventional Gaussian distribution-based likelihood function. Fig. 3.13 shows the error comparison result of the CPF-based localization system and the proposed localization system. One

remarkable point is that, occasionally, the lateral error of the CPF-based localization system increased significantly. These large lateral errors are caused by the failure in map-image matching process due to the measurement ambiguity. As we can see from the result, in case of the proposed algorithm, the lateral errors do not exceed 0.4m. From this result, we can see that the proposed algorithm effectively solves the measurement ambiguity problem and provides accurate and reliable lateral localization result.

3.5.4 Effect of the Measurement Error Model

Table 3.4 The localization performance for various standard deviation values of the Gaussian filter.

STDV [pixel]	0	11	23	35	47
Longitudinal Error [m]	1.578	1.444	1.427	1.435	1.451
Lateral Error [m]	0.742	0.710	0.069	0.708	0.721

The Gaussian noise was applied to the image to handle the variations in the pitch and roll angles. In order to see the effect of the Gaussian noise, experiments were conducted by varying the standard deviation of the Gaussian filter. Table 3.4 shows the result of the experiment. From the results, we can know that an optimal standard deviation value of the Gaussian filter exists. In our experiment, when the standard deviation was set to 23, the localization system shows the best result.

3.6 Summary

This chapter proposed a precise vehicle localization algorithm with poor feature information based on the fusion of a high-precision lane-level road map, a low-cost GPS,

in-vehicle network sensors, and a monocular camera. First, the road map presented in chapter 2 was augmented for including the dashed lane marking information and other road marking information. The augmented road map and other low-cost sensor information were integrated by a particle filter with probabilistic measurement association. The main contribution of this chapter can be summarized as follows:

- The proposed localization algorithm was designed to provide accurate vehicle pose information under the environments where feature information for localization is poor, e.g., highway. The localization system primarily uses the lane marking information, and utilizes sparsely existing road marking information.

- The localization system operates based on low-cost sensors including a low-cost GPS, in-vehicle network sensors, and a monocular camera.

- In order to handle the measurement ambiguity problem due to the poor feature information and use of low-cost sensor configuration, a modified likelihood function was proposed. From the experiment, it is demonstrated that the proposed localization system provides precise vehicle pose information for automated driving.

Chapter 4

An Application of Precise Vehicle Localization: Traffic Flow Enhancement Through the Sharing of Accurate Position Information Among Vehicles

Because of its adverse effects on economy and environment, there have been extensive efforts to resolve the traffic jam problem. However, in spite of the lots of research efforts, the traffic jam is still has considered as a serious transportation problem. In recent years, with the development of communication technologies, a number of studies have attempted to improve the traffic flow through the information exchange among vehicles or provision of useful information about road conditions to vehicles.

Platooning is one of the representative approaches for traffic flow enhancement. In the platooning system, vehicles drive by forming a small group named as the platoon for the purpose of increasing the road capacity that is defined as the maximum traffic flow which is obtainable on a given roadway [69,70]. In a platoon, since vehicles

maintain very short inter-vehicle distance, the road capacity is increased. For a platooning, vehicles are automatically controlled by their own controllers which use the information from the other vehicles within the platoon such as positions, velocities and accelerations. As shown in [71], the road capacity increases by a factor of four in the case where the average platoon size is 20 (number of group vhehicles), it is obvious that the platooning can significantly improve the traffic flow. However, for platooning, since not only all vehicles should be equipped with the automatic controllers work on a same algorithm but also complete safety should be secured, it is not realistic for the present.

More realistic approach is a passive method that assists drivers in behaving more intelligent by providing various useful information through communications. For example, providing a congestion degree of roads helps a driver can choose the optimal route. Another example is providing information about conditions of the roads and position of adjacent vehicles to drivers for the purpose of helping drivers' intelligent behaviors. We can intuitively think that the extension of drivers' view will make them driving more intelligently by anticipating the motions of other vehicles, and thus the overall traffic flow can be improved. However, it is difficult to analyze the effect of driver's behavior on the overall traffic flow. The complexity of the traffic flow makes the analysis difficult.

Instead of an analytical way, through a numerical experiment, I show that, as drivers see more headway of preceding vehicles, the overall traffic flow is improved and the traffic jam is suppressed. I assume that each vehicle c_i can know its exact global position using its own localization system, and transmit it to the following vehicles through a V2V (Vehicle to Vehicle) or V2II (Vehicle-to-Infrastructure) communications. The issues with respect to the communication methods and protocols

are out of the scope of this dissertation. Numerical experiments are performed based on the SOV (stochastic optimal velocity) model [72] that belongs to the category of the cellular automata traffic model [73]. I extend the SOV model considering the extension of drivers' view and compare the traffic flow observed on this extended-SOV (ESOV) model with the traffic flow observed on the original SOV model. The experimental result shows that the sharing of position information among vehicles can significantly improves the traffic flow and suppresses the traffic jam.

Eventually, The contribution of this chapter is summarized as follows:

- I propose an traffic model extended from the SOV model by considering the sharing of accurate position information among vehicles, which allows drivers to know the multiple headways of preceding vehicles.

- Based on the ESOV model, I examine the effect of the position information sharing these information on the traffic flow.

- Consequently, I verify that the sharing of the position information makes the traffic flow more smooth.

This chapter is organized as follows: in section 4.1, the Extended SOV model is proposed after the SOV model is explained with basic simulation results. In section 4.2, the results of the numerical experiments are presented and discussed. Finally, the summary is presented in section 4.3.

4.1 Extended SOV Model

In this section, I present an extended SOV model (ESOV) to examine the effect of accurate vehicle localization and sharing of the location information among the

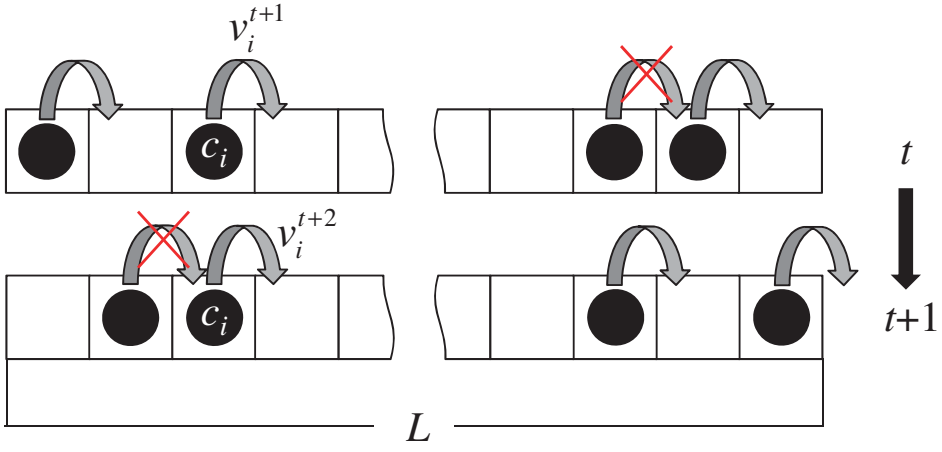


Figure 4.1 The SOV model. A vehicle i hop to the next cell with the probability v_i^{t+1} at time t. Vehicles whose next cell is occupied do not hop.

vehicles. First the original SOV model is explained in section 4.1.1, and then the extended model is explained in section 4.1.2.

4.1.1 SOV Model

This subsection describes the SOV model and presents the reason why I adopted the model. In the SOV model, N vehicles move on a single-lane which is divided into L cells as shown in Fig. 4.1. Each cell contains one vehicle at most, and collision is thus avoided. The positions of vehicles are updated in parallel at each step, i.e., all vehicles attempt to move forward. The velocity v_i^{t+1} of vehicle i at step t is defined by

$$v_i^{t+1} = (1-\alpha)v_i^t + \alpha V_i(\Delta_i), \tag{4.1}$$

where $\alpha(0 \leq \alpha \leq 1)$ is a parameter, Δ_i is the headway, and function V_i is the optimal velocity function, which is defined as

$$V_i(\Delta_i) = \frac{\tanh(\Delta_i - \beta) + \tanh\beta}{1 + \tanh\beta}. \tag{4.2}$$

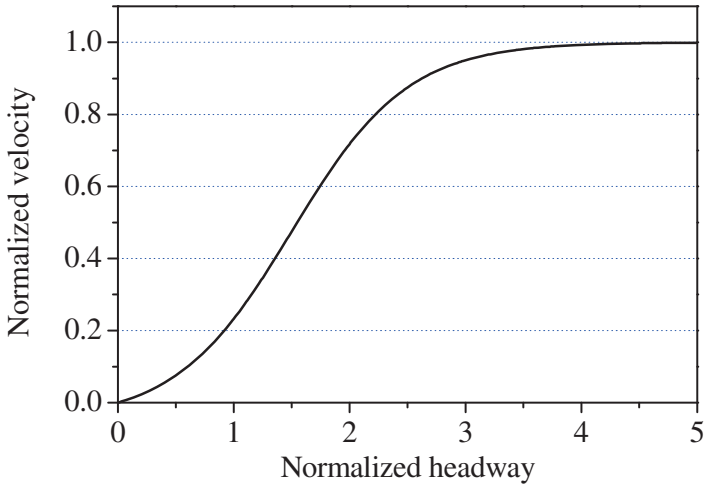


Figure 4.2 The optimal velocity function. The optimal velocity is 0 if the headway is 0 and converges to 1 as the headway increases.

Here, β is a parameter. The function V_i is plotted in Fig. 4.2, where $\beta = 1.5$. V_i reflect the reaction of the vehicle to the headway. Since V is restricted to $[0, 1]$, v_i^t is also restricted to $[0, 1]$, if v_i^0 is less than 1. Note that the normalized velocity v_i^t expresses the hop probability in this model. At each step, each vehicle hop to the next cell with the probability v_i^{t+1} except the case where the next cell is occupied. This is the reason why this model is names as stochastic OV model. Since v_i^{t+1} is the average of v_i^t and V_i, from the Eq. (4.1), variation of the hopping probability of a vehicle depends on its headway.

In spite of its simplicity, the SOV model is plausible for tracing the real traffic flow features. Fig. 4.3 shows the fundamental diagrams of the SOV model with $\alpha = 0.01$ in Eq. (4.1) and $\beta = 1.5$ in Eq. (4.2) for four iteration numbers. In our simulation, the length of the road L was set to 1000 cells and a cyclic boundary condition is applied, i.e., the last cell is linked to the first cell. At the initial step $t = 0$, vehi-

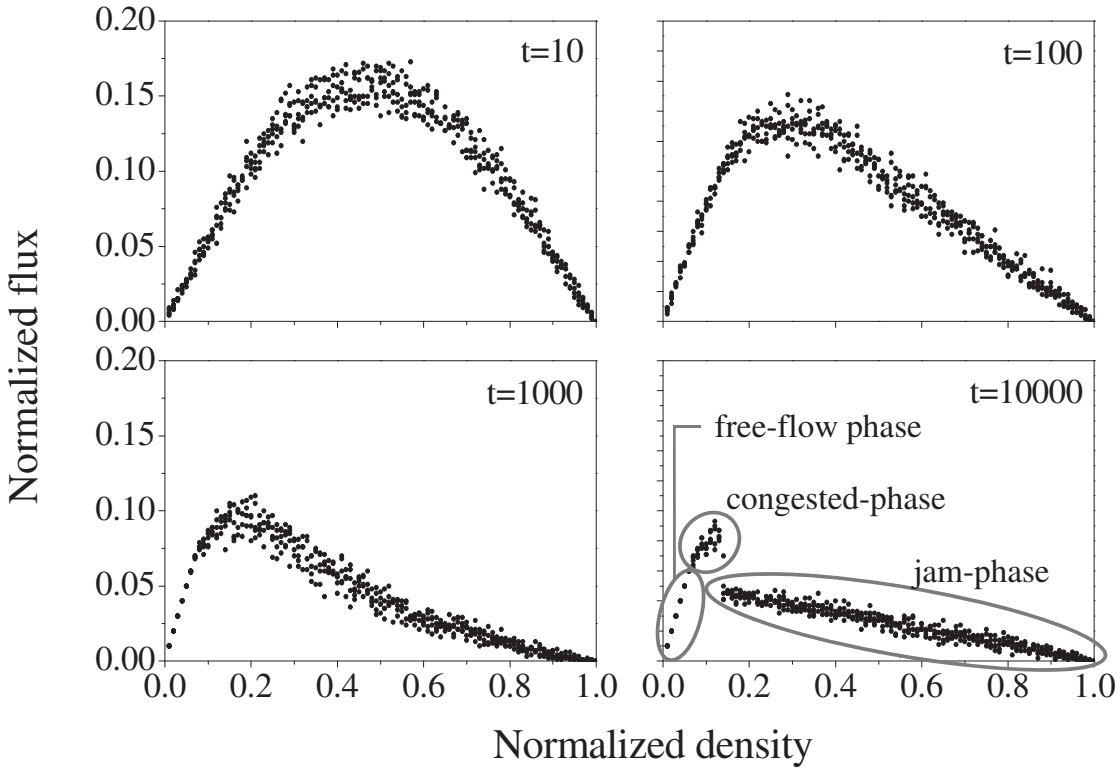


Figure 4.3 The fundamental diagram of the SOV model for $t = 10, 100, 1000$ and 10000. The parameters are $\alpha = 0.01$ and $\beta = 1.5$, and the length of the road is L.

cles are uniformly distributed and velocities are set to $v_i^0 = 0$. The cyclic boundary is appropriate for presenting the homogeneous road which has no bottlenecks. The fundamental diagram shows the variation of the flux according to the density, where the flux is defined by the product of the velocity and density, and thus presents the number of vehicles that pass the unit length per unit time. As time passed, the flux begins to decrease, and finally, the traffic reaches to a stationary state in which the traffic jam exist.

An interesting point is that a discontinuous point appears in the fundamental diagram of stationary state ($t = 10000$) and three different phases are observed around

that point: free-flow phase, congested phase, and jam phase. In the free-flow phase, vehicles move without any interruptions. Therefore, the flux of the free-flow phase is same to the density, where the maximum velocity is 1. In the congested phase, some vehicles are clustered and the average velocity and the flux thus decrease. During the congested phase the small clusters of vehicles are coexist with the free vehicles. A traffic jam is formed in the jam phase. We can see the transition processes of phases through Fig. 4.4. Fig. 4.4-(a) and (b) show the simulation results that present the time evolution of the flux and the distribution of the vehicles at each step respectively, where the density ρ is 1.4. We can know that a traffic jam occurs at the density $\rho = 1.4$ in a stationary state from Fig. 4.3. These two diagrams show the transient process from the initial state to the stationary state. In Fig. 4.4-(a), the phase transition points are clearly appeared. Since I set the initial velocities to v_i^0=1, the traffic start at a free-flow phase. The free-flow phase is lasted until about $t = 5000$ and then replaced by the congested phase. The congested phase is broken at about $t = 10000$, and from this time, the jam phase is lasted. The spatio-temporal diagram which shown in Fig. 4.4 shows the physical features of the congested and jam phase by presenting the position of vehicles at each iteration number. The Y-axis of the diagram denotes the positions of vehicles and the X-axis means the iteration number. In other words, position of each vehicle at each step is denoted as a dot in the diagram, where vehicles move from bottom to up. As shown in the diagram, in the congested phase the vehicles form many small clusters. Relatively darker regions in the diagram present the clustered vehicles. A cluster is produced by a fluctuation of the velocity of a vehicle. If the density of vehicles of the region in which the cluster is formed is high, the cluster becomes a seed of a traffic jam. The diagram shows the forming process of the traffic jam that looks like a black band. Normally, a traffic jam is propagated backward.

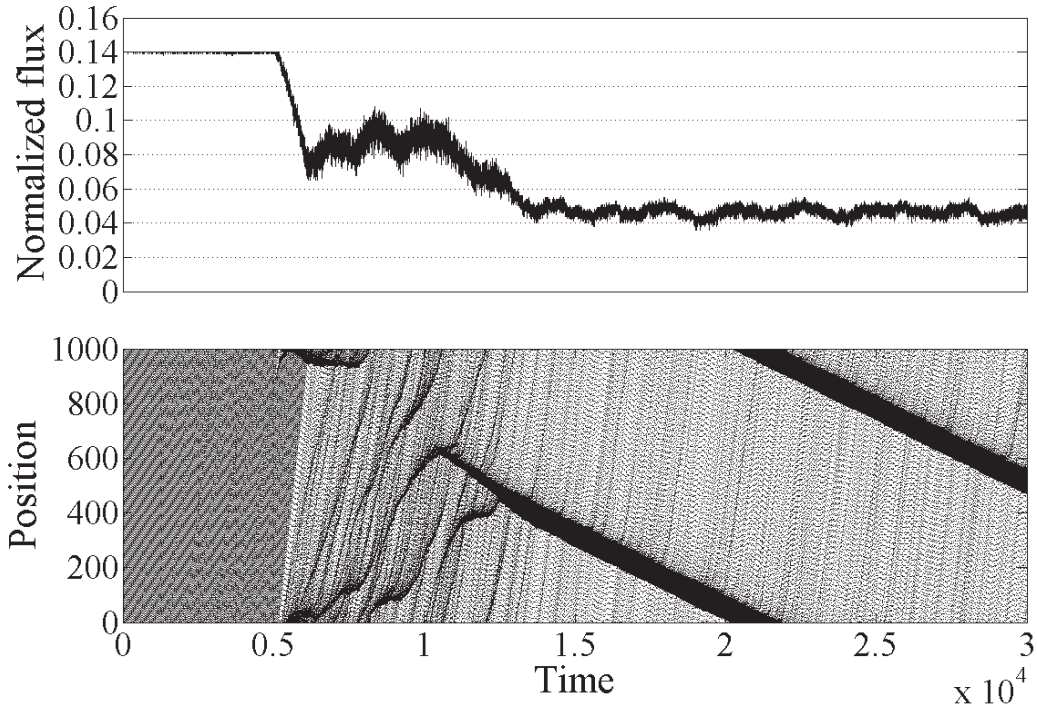


Figure 4.4 (a) The time evolution of flux at the density $\rho = 0.14$, starting from the unifrom state in which all the inter-vehicle distances and velocities of vehicles are identical. The initial velocities are set as $v_i^0 = 1$. (b) The spatio-temporal diagram of the SOV model. A vehicle is denoted as a dot.

The phase transition is important feature that is appeared in the real traffic. This point indicates that the SOV model is realistic and thus led me to adopt this model.

4.1.2 Extended SOV Model

The extended SOV model that I propose considers the intelligent behavior of drivers who receive more information about motions of preceding vehicles. Unlike the original SOV model, the extended model assume that the drivers can see headways of multiple preceding vehicles through the communications, i.e., vehicles receive the headway

or position information from preceding vehicles through the V2V or V2I communications. To reflect this so-called expansion-of-view, I modified the original model without loss of generality as follows:

$$v_i^{t+1} = (1-\alpha)v_i^t + \alpha V_i(w_1\Delta_i + w_2\Delta_{i+1} + ... + w_k\Delta_{i+k-1}), \tag{4.3}$$

where I assume that each vehicle know not only its headway but also the headways of $k-1$ preceding vehicles, and w is weighting factors that satisfies $\sum_{j=1}^{k} w_j$. The extended model is same as the original model when k=1. I give the condition $\sum_{j=1}^{k} w_j$ to make the drivers do the same behavior with the case of the original model when its headway and all the headways of $k-1$ preceding vehicles are identical, i.e., $\Delta_i = \Delta_{i+1}... = \Delta_{i+k}$. Except this condition, I do not suggest a specific rule to determine the weighting factors. I remain it a future work to investigate the effect of weighting factors on the traffic flow. In this dissertation I simply set the weighting factors so that they have same values, i.e., $w_j = 1/k$.

In this model, if the headways of preceding vehicles are relatively long, drivers increase their speed more than the case where the drivers just know its headway (the case of the original model). Similarly, if the headways of preceding vehicles are relatively short, drivers decrease their speed more. In other words, drivers anticipate the near-future motion of preceding vehicles, and they thus accelerate if the preceding vehicles are likely to accelerate even though its current headway is short. The opposite case is the same. Intuitively, we can think that this so-called anticipation effect will make the traffic flow more stable and smooth. As pointed out in [74], a traffic jam can be induced by a tiny fluctuation of the velocity of a vehicle. In the proposed model, since vehicles anticipate the motion of the preceding vehicles, they are less susceptible to the fluctuation of the immediate preceding vehicle. Hence, it is expected that the anticipation effect will suppress the traffic jam. In the next section, I show the result

of the numerical experiment to confirm this hypothesis.

4.2 Results and Discussions

In this section, I present the result of the numerical experiment. First, to ascertain whether the anticipation effect according to the extension of drivers view improves the traffic flow, I examined the effect of the number of front vehicles that each vehicle can perceive on the traffic flow. Fig. 4.5 shows the fundamental diagrams obtained from the numerical experiment based on the extended SOV model. Here, the system size is $L = 1000$ and the cyclic boundary condition was adopted. The graph shows the average flux during the time period of $t = 9000 \sim 10000$ corresponding to the stationary state. Parameters are set as $\alpha = 0.01$ and $\beta = 1.5$ and the initial velocities of vehicles are set as $v_i^0 = 0.5$ identically.

Note that, as k increase, not only the overall average flux increases, but also the formation of the traffic jam is suppressed. From the result, we can know that overall traffic flow is improved about 40% when $k = 3$ and 60% when $k = 5$ than the case where $k = 1$. We also can know that the density value at which the free-flow phase is broken is changed. Whereas the free-flow phase is broken at the density of $\rho = 0.8$ for $k = 1$, it is broken at the density of $\rho = 0.13$ for $k = 3$ and at the density of $\rho = 0.2$ for $k = 5$. Finally, the density at which the traffic jam occurs is also changed. The traffic jam is formed at $\rho = 0.17$ for $k = 1$, $\rho = 0.2$ for $k = 3$, and $\rho = 0.22$ for $k = 5$ respectively.

More detailed features can be seen from the time evolutionary flux diagrams and the spatio-temporal diagrams for $k = 1, 3$ and 5 as shown in Fig. 4.6a, 4.6b and 4.6c respectively. Here, the density is set to $\rho = 0.18$ and the initial velocities are set to $v_i^0 = 0.5$. Even though the results are obtained at the same density value,

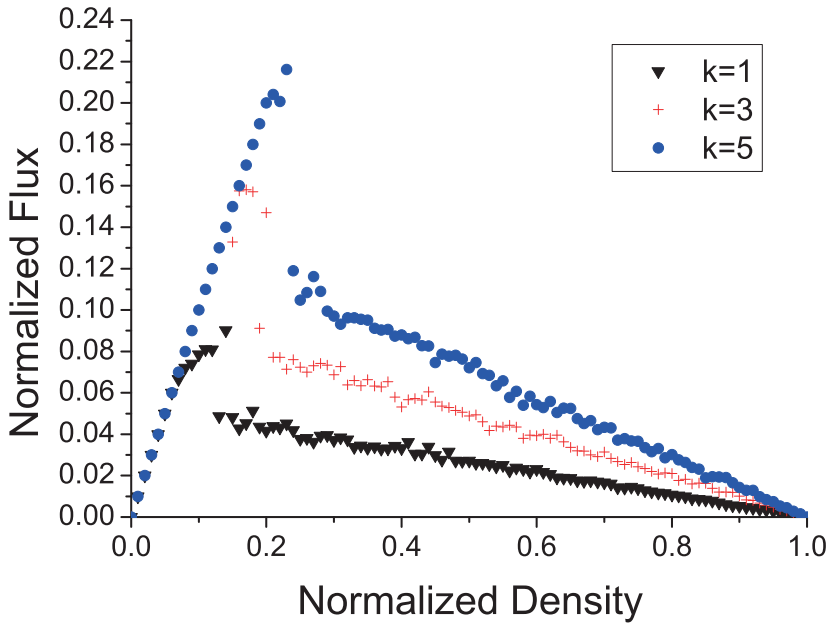
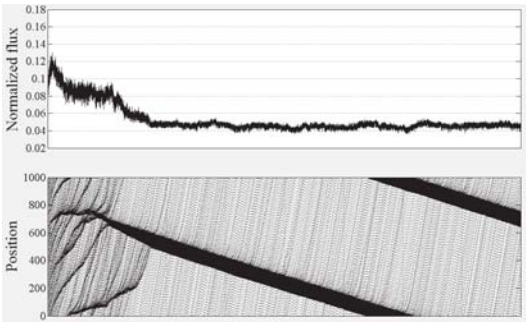
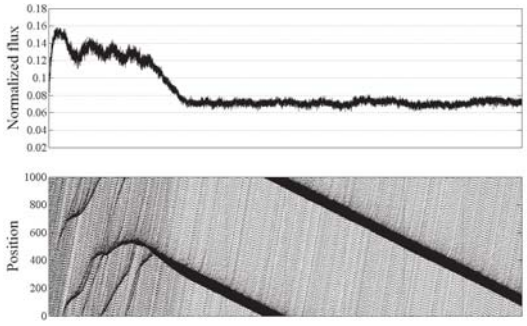


Figure 4.5 The fundamental diagram of the SOV model for $t = 10, 100, 1000$ and 10000. The parameters are $\alpha = 0.01$ and $\beta = 1.5$, and the length of the road L is 1000.

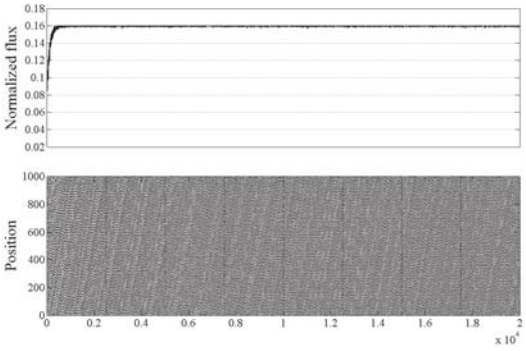
the traffic features shown in diagrams of Fig. 4.6a, Fig. 4.6b, and Fig. 4.6c are fairly different. First, the forming of the cluster is suppressed as k increase. Fewer clusters are observed in the spatio-temporal diagram when $k = 3$ than the case where $k = 1$. One remarkable point is that, for $k = 5$, no clusters and no traffic jam are observed and the traffic is always in the free-flow phase in which the flux is always maximum as shown in Fig. 4.6c. Therefore, we can see that, by the anticipation effect according to the sharing of position information among vehicles, the clusters are dissolved better before growing to the traffic jam. Second, the jam is suppressed. Comparing the spatio-temporal diagram where $k = 3$ with the diagram where $k = 1$, we can see that the width of the jam get narrower and the slope of the jam propagation get steeper as k increases. The narrow width and steep slope of traffic jam means that a vehicle



(a) $k = 1$



(b) $k = 3$



(c) $k = 5$

Figure 4.6 The time evolutionary flux diagrams and the spatio-temporal diagrams. The density is $\rho = 1.8$ and the number of headway that drivers can see are (a) $k = 1$, (b) $k = 3$ and (c) $k = 5$.

that entered the jam area can escape the area more quickly.

4.3 Summary

This chapter demonstrated that the sharing of accurate vehicle position among vehicle can improve the traffic flow and suppress the traffic jam. In order to evaluate the effect of the position information sharing, an ESOV traffic model is proposed by modifying the conventional SOV traffic model. The result of the numerical experiment based

on the ESOV model demonstrates that the traffic flow is increased as a driver can percept the positions of more preceding vehicles.

Chapter 5

Conclusion

In this dissertation I have studied on the mapping and localization problem for automated driving. The accuracy and cost are two primary requirements of the mapping and localization for automated driving. In chapter 2, a road map generation system to generate precise and efficient lane-level road map was proposed. The road map generation system integrates a 3D LIDAR data and high-precision vehicle positioning system to acquire accurate road geometry data. Acquired road geometry data is represented as sets of piecewise polynomial curves in order to increase the storage efficiency and the usability. From extensive experiments based on a real urban and highway road data, I verified that proposed road map generation system generate a road map that is accurate and more efficient than previous road maps in terms of the storage efficiency and usability.

In chapter 3, a precise vehicle localization system was proposed based on the lane-level road map presented in chapter 2, and low-cost sensors. A measurement ambiguity problem due to the use of low-cost sensors and poor feature information was presented,

and a probabilistic measurement association-based particle filter was proposed to resolve the measurement ambiguity problem. Through extensive experiment based on real highway data, the accuracy and reliability of the localization system was evaluated.

In chapter 4, an application of the accurate vehicle localization system was presented. I demonstrated that sharing of accurate position information among vehicles can improve the traffic flow and suppress the traffic jam effectively. The effect of the position information sharing was evaluated based on numerical experiments. For this, the ESOV traffic model was proposed by extending conventional SOV traffic model. The numerical experiments show that the traffic flow is increased as a driver can perceive the position of more preceding vehicles.

I hope this study will contribute to the future of the automated vehicle systems and will be a good foundation for follow-up studies. This study is likely to be expanded in various aspects. The road map generation system proposed in this dissertation operates in a semi-automatic manner. It will be one of good research topics to automatize the road map generation process. The automation of the map generation process is essential to save the cost and time which are needed for replacing current road maps with the map for automated driving. In addition, only the lane information is contained in the map in this dissertation, however, the map will become more valuable if the other additional information is contained in the map such as traffic lights, road signs, and so on. In conjunction with this, the proposed localization system only uses road marking information in the map. However, other landmark information contained in the map can be used to increase the accuracy and reliability of the localization system. I hope many follow-up studies will be carried out and many valuable research results will be produced.

Bibliography

[1] K. Seong-Woo, W. Liu, M. H. Ang, E. Frazzoli, and D. Rus, "The impact of cooperative perception on decision making and planning of autonomous vehicles," *Intelligent Transportation Systems Magazine, IEEE*, vol. 7, no. 3, pp. 39–50, 2015.

[2] C. Urmson, J. Anhalt, D. Bagnell, C. Baker, R. Bittner, M. Clark, J. Dolan, D. Duggins, T. Galatali, C. Geyer *et al.*, "Autonomous driving in urban environments: Boss and the urban challenge," *Journal of Field Robotics*, vol. 25, no. 8, pp. 425–466, 2008.

[3] K. Chu, M. Lee, and M. Sunwoo, "Local path planning for off-road autonomous driving with avoidance of static obstacles," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 13, no. 4, pp. 1599–1616, Dec 2012.

[4] J. Naranjo, C. Gonzalez, R. Garcia, and T. de Pedro, "Lane-change fuzzy control in autonomous vehicles for the overtaking maneuver," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 9, no. 3, pp. 438–450, Sept 2008.

[5] R. Toledo-Moreo, D. Bétaille, F. Peyret, and J. Laneurit, "Fusing GNSS, dead-reckoning, and enhanced maps for road vehicle lane-level navigation," *Selected*

Topics in Signal Processing, IEEE Journal of, vol. 3, no. 5, pp. 798–809, Oct 2009.

[6] R. Toledo-Moreo, D. Bétaille, and F. Peyret, "Lane-level integrity provision for navigation and map matching with GNSS, dead reckoning, and enhanced maps," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 11, no. 1, pp. 100–112, March 2010.

[7] Z. Tao, P. Bonnifait, V. Fremont, and J. Ibanez-Guzman, "Mapping and localization using gps, lane markings and proprioceptive sensors," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on.* IEEE, 2013, pp. 406–412.

[8] N. Suganuma and T. Uozumi, "Precise position estimation of autonomous vehicle based on map-matching," in *Intelligent Vehicles Symposium (IV), 2011 IEEE*, June 2011, pp. 296–301.

[9] M. Schreiber, C. Knoppel, and U. Franke, "Laneloc: Lane marking based localization using highly accurate maps," in *Intelligent Vehicles Symposium (IV), 2013 IEEE*, June 2013, pp. 449–454.

[10] A. Schindler, "Vehicle self-localization with high-precision digital maps," in *Intelligent Vehicles Symposium (IV), 2013 IEEE*, June 2013, pp. 141–146.

[11] E. Consortium *et al.*, "Enhanced digital mapping project. final report." EDMap Project eSafety Forum.(2005). Digital maps Working Group Final Report. European Commission (eSafety Forum), Brussels, 2004.

[12] D. Bétaille and R. Toledo-Moreo, "Creating enhanced maps for lane-level vehicle navigation," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 11, no. 4, pp. 786–798, Dec 2010.

[13] B. Schwarz, "Lidar: Mapping the world in 3D," *Nature Photonics*, vol. 4, no. 7, pp. 429–430, 2010.

[14] K. Jo and M. Sunwoo, "Generation of a precise roadway map for autonomous cars," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 15, no. 3, pp. 925–937, June 2014.

[15] K. Ichida, T. Kiyono, and F. Yoshimoto, "Curve fitting by a one-pass method with a piecewise cubic polynomial," *ACM Transactions on Mathematical Software (TOMS)*, vol. 3, no. 2, pp. 164–174, 1977.

[16] M. Plass and M. Stone, "Curve-fitting with piecewise parametric cubics," in *ACM SIGGRAPH Computer Graphics*, vol. 17, no. 3. ACM, 1983, pp. 229–239.

[17] M. Haklay and P. Weber, "Openstreetmap: User-generated street maps," *Pervasive Computing, IEEE*, vol. 7, no. 4, pp. 12–18, 2008.

[18] J. Ziegler, P. Bender, M. Schreiber, H. Lategahn, T. Strauss, C. Stiller, T. Dang, U. Franke, N. Appenrodt, C. G. Keller *et al.*, "Making bertha drive—an autonomous journey on a historic route," *Intelligent Transportation Systems Magazine, IEEE*, vol. 6, no. 2, pp. 8–20, 2014.

[19] A. Joshi and M. R. James, "Generation of accurate lane-level maps from coarse prior maps and lidar," *Intelligent Transportation Systems Magazine, IEEE*, vol. 7, no. 1, pp. 19–29, 2015.

[20] Y.-W. Seo, C. Urmson, and D. Wettergreen, "Exploiting publicly available cartographic resources for aerial image analysis," in *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*, ser. SIGSPATIAL '12. New York, NY, USA: ACM, 2012, pp. 109–118.

[21] J. Hu, A. Razdan, J. Femiani, M. Cui, and P. Wonka, "Road network extraction and intersection detection from aerial images by tracking road footprints," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 45, no. 12, pp. 4144–4157, Dec 2007.

[22] M. Amo, F. Martinez, and M. Torre, "Road extraction from aerial images using a region competition algorithm," *Image Processing, IEEE Transactions on*, vol. 15, no. 5, pp. 1192–1201, May 2006.

[23] S. Rogers, "Creating and evaluating highly accurate maps with probe vehicles," in *Intelligent Transportation Systems, 2000. Proceedings. 2000 IEEE*, 2000, pp. 125–130.

[24] A. Schindler, G. Maier, and F. Janda, "Generation of high precision digital maps using circular arc splines," in *Intelligent Vehicles Symposium (IV), 2012 IEEE*, June 2012, pp. 246–251.

[25] S. Schroedl, K. Wagstaff, S. Rogers, P. Langley, and C. Wilson, "Mining GPS traces for map refinement," *Data mining and knowledge Discovery*, vol. 9, no. 1, pp. 59–87, 2004.

[26] F. Caron, E. Duflos, D. Pomorski, and P. Vanheeghe, "GPS/IMU data fusion using multisensor Kalman filtering: introduction of contextual aspects," *Information Fusion*, vol. 7, no. 2, pp. 221 – 230, 2006.

[27] H. Qi and J. Moore, "Direct Kalman filtering approach for GPS/INS integration," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 38, no. 2, pp. 687–693, Apr 2002.

[28] X. Chen, B. Kohlmeyer, M. Stroila, N. Alwar, R. Wang, and J. Bach, "Next generation map making: geo-referenced ground-level lidar point clouds for automatic retro-reflective road feature extraction," in *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 2009, pp. 488–491.

[29] A. Boyko and T. Funkhouser, "Extracting roads from dense point clouds in large scale urban environment," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 66, no. 6, pp. S2–S12, 2011.

[30] A. Mancini, E. Frontoni, and P. Zingaretti, "Automatic road object extraction from mobile mapping systems," in *Mechatronics and Embedded Systems and Applications (MESA), 2012 IEEE/ASME International Conference on*. IEEE, 2012, pp. 281–286.

[31] H. Guan, J. Li, Y. Yu, C. Wang, M. Chapman, and B. Yang, "Using mobile laser scanning data for automated extraction of road markings," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 87, pp. 93–107, 2014.

[32] S. Brummer, F. Janda, G. Maier, and A. Schindler, "Evaluation of a mapping strategy based on smooth arc splines for different road types," in *Intelligent Transportation Systems - (ITSC), 2013 16th International IEEE Conference on*, Oct 2013, pp. 160–165.

[33] A. Schindler, G. Maier, and S. Pangerl, "Exploiting arc splines for digital maps," in *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, Oct 2011, pp. 1–6.

[34] J. Goldbeck and B. Huertgen, "Lane detection and tracking by video sensors," in *Intelligent Transportation Systems, 1999. Proceedings. 1999 IEEE/IEEJ/JSAI International Conference on*, 1999, pp. 74–79.

[35] E. Safra, Y. Kanza, Y. Sagiv, and Y. Doytsher, "Efficient integration of road maps," in *Proceedings of the 14th annual ACM international symposium on Advances in geographic information systems*. ACM, 2006, pp. 59–66.

[36] A. Vatavu, R. Danescu, and S. Nedevschi, "Environment perception using dynamic polylines and particle based occupancy grids," in *Intelligent Computer Communication and Processing (ICCP), 2011 IEEE International Conference on*. IEEE, 2011, pp. 239–244.

[37] S. Sivaraman and M. M. Trivedi, "Dynamic probabilistic drivability maps for lane change and merge driver assistance," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 15, no. 5, pp. 2063–2073, 2014.

[38] V. Gikas and J. Stratakos, "A novel geodetic engineering method for accurate and automated road/railway centerline geometry extraction based on the bearing diagram and fractal behavior," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 13, no. 1, pp. 115–126, March 2012.

[39] H. Park and J.-H. Lee, "B-spline curve fitting based on adaptive curve refinement using dominant points," *Computer-Aided Design*, vol. 39, no. 6, pp. 439 – 451, 2007.

[40] P. Sherar, "Variational based analysis and modelling using B-splines," Ph.D. dissertation, Cranfield University, 2004.

[41] D.-S. Kim, J. Ryu, H.-C. Lee, and H. Shin, "The conversion of a dynamic b-spline curve into piecewise polynomials in power form," *Computer-Aided Design*, vol. 34, no. 4, pp. 337–345, 2002.

[42] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, p. 0278364913491297, 2013.

[43] L. Soomok, K. Seong-Woo, and S. Seung-Woo, "Accurate ego-lane recognition utilizing multiple road characteristics in a bayesian network framework," in *Intelligent Vehicles Symposium (IV), 2015 IEEE*, June 2015, pp. 543–548.

[44] K. Seung-Nam, L. Soomok, H. Junhwa, and S. Seung-Woo, "Multi-lane detection based on accurate geometric lane estimation in highway scenarios," in *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*. IEEE, 2014, pp. 221–226.

[45] H. Junhwa, K. Seung-Nam, and S. Seung-Woo, "Multi-lane detection in urban driving environments using conditional random fields," in *Intelligent Vehicles Symposium (IV), 2013 IEEE*. IEEE, 2013, pp. 1297–1302.

[46] Y. Zhou and A. Toga, "Efficient skeletonization of volumetric objects," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 5, no. 3, pp. 196–209, Jul 1999.

[47] H. Wang, J. Kearney, and K. Atkinson, "Arc-length parameterized spline curves for real-time simulation," *Curve and surface design: Saint-Malo 2002.*, pp. 387–396, 2003.

[48] A. Aldoma, Z.-C. Marton, F. Tombari, W. Wohlkinger, C. Potthast, B. Zeisl, R. B. Rusu, S. Gedikli, and M. Vincze, "Point cloud library," *IEEE Robotics & Automation Magazine*, vol. 1070, no. 9932/12, 2012.

[49] R. Toledo-Moreo, M. Zamora-Izquierdo, B. Úbeda-Miñarro, A. F. Gómez-Skarmeta *et al.*, "High-integrity imm-ekf-based road vehicle navigation with low-cost gps/sbas/ins," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 8, no. 3, pp. 491–511, 2007.

[50] S. Rezaei and R. Sengupta, "Kalman filter-based integration of dgps and vehicle sensors for localization," *Control Systems Technology, IEEE Transactions on*, vol. 15, no. 6, pp. 1080–1088, 2007.

[51] J. Huang and H.-S. Tan, "A low-order dgps-based vehicle positioning system under urban environment," *Mechatronics, IEEE/ASME Transactions on*, vol. 11, no. 5, pp. 567–575, 2006.

[52] Y. Cui and S. S. Ge, "Autonomous vehicle positioning with gps in urban canyon environments," *Robotics and Automation, IEEE Transactions on*, vol. 19, no. 1, pp. 15–25, 2003.

[53] Z. Tao, P. Bonnifait, V. Fremont, and J. Ibanez-Guzman, "Lane marking aided vehicle localization," in *Intelligent Transportation Systems-(ITSC), 2013 16th International IEEE Conference on*. IEEE, 2013, pp. 1509–1515.

[54] T. Wu and A. Ranganathan, "Vehicle localization using road markings," in *Intelligent Vehicles Symposium (IV), 2013 IEEE*. IEEE, 2013, pp. 1185–1190.

[55] H. Lategahn and C. Stiller, "Vision-only localization," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 15, no. 3, pp. 1246–1257, 2014.

[56] M. Schreiber, C. Knoppel, and U. Franke, "Laneloc: Lane marking based localization using highly accurate maps," in *Intelligent Vehicles Symposium (IV), 2013 IEEE*. IEEE, 2013, pp. 449–454.

[57] J. Ziegler, H. Lategahn, M. Schreiber, C. G. Keller, C. Knoppel, J. Hipp, M. Haueis, and C. Stiller, "Video based localization for bertha," in *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*. IEEE, 2014, pp. 1231–1238.

[58] K. Jo, Y. Jo, J. K. Suhr, H. G. Jung, and M. Sunwoo, "Precise localization of an autonomous car based on probabilistic noise models of road surface marker features using multiple cameras," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 16, no. 6, pp. 3377–3392, 2015.

[59] K. Jo, K. Chu, and M. Sunwoo, "Interacting multiple model filter-based sensor fusion of gps with in-vehicle sensors for real-time vehicle positioning," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 13, no. 1, pp. 329–343, 2012.

[60] A. Geiger, J. Ziegler, and C. Stiller, "Stereoscan: Dense 3d reconstruction in real-time," in *Intelligent Vehicles Symposium (IV)*, 2011.

[61] S. Thrun and J. J. Leonard, "Simultaneous localization and mapping," in *Springer handbook of robotics*. Springer, 2008, pp. 871–889.

[62] M. Montemerlo and S. Thrun, "Simultaneous localization and mapping with unknown data association using fastslam," in *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, vol. 2. IEEE, 2003, pp. 1985–1991.

[63] M. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba, "A solution to the simultaneous localization and map building (slam) problem,"

Robotics and Automation, IEEE Transactions on, vol. 17, no. 3, pp. 229–241, 2001.

[64] D. Gruyer, R. Belaroussi, and M. Revilloud, "Map-aided localization with lateral perception," in *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*. IEEE, 2014, pp. 674–680.

[65] N. Suganuma and T. Uozumi, "Precise position estimation of autonomous vehicle based on map-matching," in *Intelligent Vehicles Symposium (IV), 2011 IEEE*. IEEE, 2011, pp. 296–301.

[66] J. Levinson and S. Thrun, "Robust vehicle localization in urban environments using probabilistic maps," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. IEEE, 2010, pp. 4372–4378.

[67] N. Mattern and G. Wanielik, "Camera-based vehicle localization at intersections using detailed digital maps," in *Position Location and Navigation Symposium (PLANS), 2010 IEEE/ION*. IEEE, 2010, pp. 1100–1107.

[68] N. Mattern, R. Schubert, and G. Wanielik, "High-accurate vehicle localization using digital maps and coherency images," in *Intelligent Vehicles Symposium (IV), 2010 IEEE*. IEEE, 2010, pp. 462–469.

[69] D. Swaroop, "String stability of interconnected systems: An application to platooning in automated highway systems," *California Partners for Advanced Transit and Highways (PATH)*, 1997.

[70] D. Swaroop and J. Hedrick, "Constant spacing strategies for platooning in automated highway systems," *Journal of dynamic systems, measurement, and control*, vol. 121, no. 3, pp. 462–470, 1999.

[71] U. Karaaslan, P. Varaiya, and J. Walrand, "Two proposals to improve freeway traffic flow," in *American Control Conference, 1991.* IEEE, 1991, pp. 2539–2544.

[72] M. Kanai, K. Nishinari, and T. Tokihiro, "Stochastic optimal velocity model and its long-lived metastability," *Physical Review E*, vol. 72, no. 3, p. 035102, 2005.

[73] K. Nagel and M. Schreckenberg, "A cellular automaton model for freeway traffic," *Journal de physique I*, vol. 2, no. 12, pp. 2221–2229, 1992.

[74] Y. Sugiyama, M. Fukui, M. Kikuchi, K. Hasebe, A. Nakayama, K. Nishinari, S.-i. Tadaki, and S. Yukawa, "Traffic jams without bottlenecks—experimental evidence for the physical mechanism of the formation of a jam," *New Journal of Physics*, vol. 10, no. 3, p. 033001, 2008.

국문 초록

본 논문은 무인자율주행을 위한 정밀하고 효율적인 지도 생성 및 측위 알고리즘을 제안하는 것을 목적으로 한다. 지도 생성 및 측위는 무인자율주행을 위한 핵심 요소로서, 이를 위한 연구에서 주요하게 고려해야할 점은 정확도 및 정밀도를 최대화 하면서 비용은 최소화 하는 것이다. 이러한 목표를 위해, 본 논문은 정밀하면서 비용 효율적인 차선 수준의 도로 지도를 생성하기 위한 시스템을 제안하고, 생성한 차로 지도와 저가형 센서를 활용한 고정밀 측위 시스템을 제안한다.

2 장에서는 도로 지도 생성 시스템을 제시한다. 제안하는 도로 지도 생성 시스템은 먼저 3 차원 라이다 데이터와 고정밀 측위 장비를 결합하여 정확한 도로 지오메트리 데이터를 취득한다. 다음으로, 도로 지도의 저장공간 효율성 및 활용성을 높이기 위해 취득한 도로 지오메트리 데이터를 구간적 다항식 형태의 곡선 식으로 모델링한다. 실제 도심 및 고속도로 데이터를 이용한 실험을 통해 제안한 도로 생성 시스템이 기존 방식에 비해 저장공간 효율성 및 활용성 측면에서 뛰어난 효율성을 지님을 보인다.

3 장에서는 2 장에서 생성한 도로 지도 및 다양한 저가형 센서를 활용한 자동차 측위 시스템을 제시한다. 제안하는 측위 시스템은 피쳐 정보가 부족하여 기존 측위 기술을 적용하기 어려운 환경에서의 정확한 측위를 목표로 한다. 저가형 센서의 사용 및 부족한 피쳐 정보로 인해 발생하는 이른바 계측값 모호성 문제를 제시하고, 이 문제를 해결하기 위한 확률적 계측값 연계 기반의 입자 필터를 제안한다. 실제 도로 데이터를 이용한 실험을 통해 제안한 측위 시스템의 정확도 및 신뢰도를 평가한다.

4 장에서는 정확한 자동차 측위의 한 응용으로서, 정확한 측위 데이터가 교통 흐름 개선에 기여할 수 있음을 보인다. 차량 간 측위 데이터 교환의 효과를 수치적 실험을 통해 입증하기 위해 기존에 널리 사용되는 SOV 교통 모델에 차량 간 측위 데이터 교환 효과를

추가한 확장 SOV 교통 모델을 제안한다. 제안한 모델을 이용하여 차량 간에 정확한 측위 데이터를 교환함으로써 교통 흐름을 크게 개선할 수 있음을 보인다.

주요어: 지능형 자동차, 도로 지도 생성, 자동차 측위

학번: 2009-20753